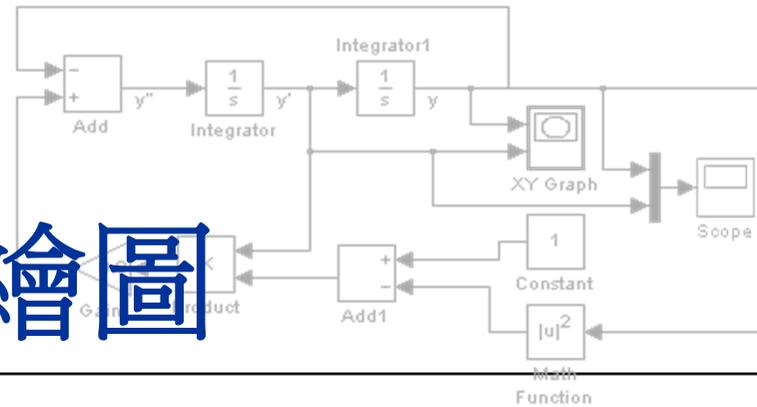


二維平面繪圖

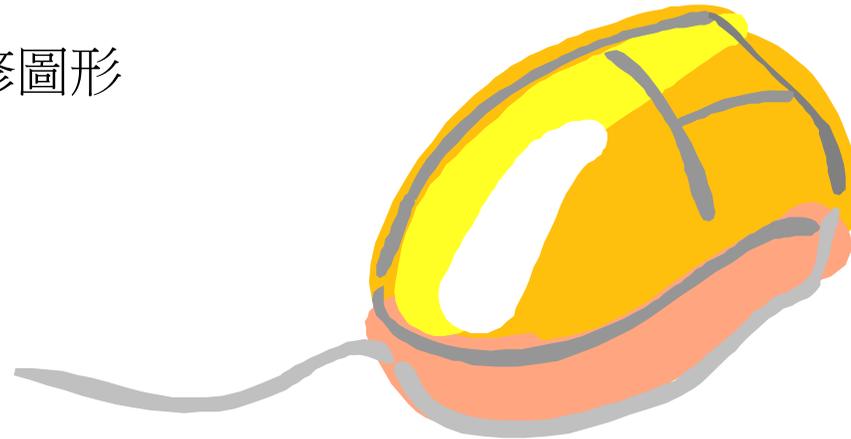


Matlab的基本繪圖函數plot

編修函數的圖形

fplot與ezplot函數

利用Property editor來編修圖形





基本繪圖函數

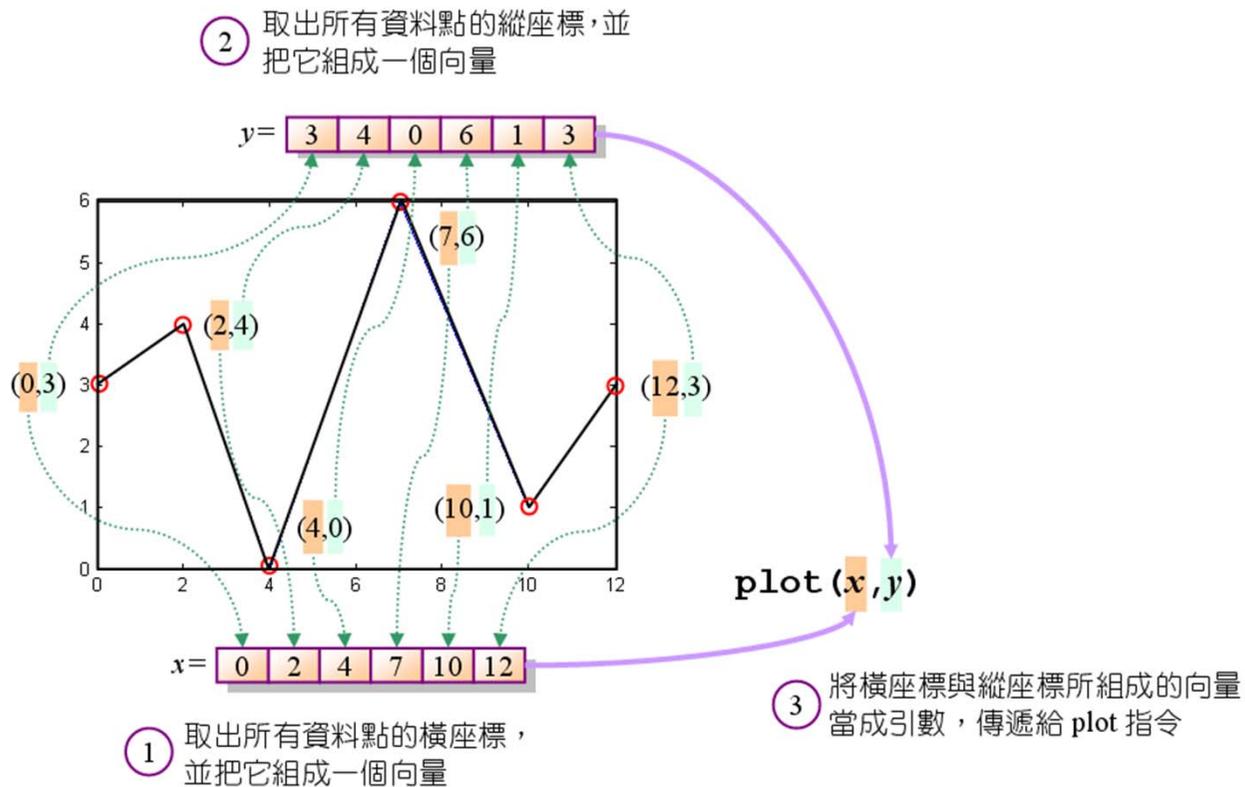
基本的二維繪圖函數

- 在Matlab裡，最常使用的二維繪圖函數是plot函數：

表 4.1.1 plot 函數的使用

函 數	說 明
<code>plot(x,y)</code>	以 x 為資料點的橫座標所組成的向量， y 為縱座標所組成的向量，描點繪出 (x,y) 的曲線圖
<code>plot(y)</code>	x 的間距為 1，描點繪出 (x,y) 的曲線圖

- 下圖是由資料點擷取出橫座標與縱座標的示意圖：



二維圖形的編修

- 利用下面的語法可編修二維函數的圖形：

表 4.1.2 修飾 plot 函數所繪出的圖形

函 數	說 明
<code>plot(x, y, 'str')</code>	以字串 <i>str</i> 所指定的格式繪出二維圖形
<code>plot(x₁, y₁, 'str₁', x₂, y₂, 'str₂', ...)</code>	以字串 <i>str₁</i> 所指定的格式繪出 (<i>x₁</i> , <i>y₁</i>) 的圖形，以 <i>str₂</i> 所指定的格式繪出 (<i>x₂</i> , <i>y₂</i>) 的圖形，以此類推
<code>plot(x₁, y₁, 'str', 'p_str', property, ...)</code>	根據繪圖性質 <i>p_str</i> 來繪圖，其中 <i>p_str</i> 可為： <i>LineWidth</i> — 設定線條寬度 <i>MarkerFaceColor</i> — 設定標記的顏色 <i>MarkerEdgeColor</i> — 設定標記的邊框顏色 <i>MarkerSize</i> — 設定標記的大小

- 
-
- 下表列出了字串`str`可使用的控制碼，以及它們所代表的意義：

表 4.1.3 `plot` 函數的控制碼（一），控制資料點的顯示符號

符號	說明	符號	說明
.	繪點	^	繪出「^」符號
*	繪出星號	v	繪出「v」符號（小寫 v）
o	繪出小圓（小寫字母 o）	s 或 square	繪出正方形
+	繪出加號	d 或 diamond	繪出菱形
x	繪出打叉符號（小寫字母 x）	p 或 pentagram	繪出五角形
<	繪出「<」符號	h 或 hexagram	繪出六角形
>	繪出「>」符號	none	不繪出任何形狀（預設）



表 4.1.4 plot 函數的控制碼（二），控制線條樣式

線條樣式	說明	線條樣式	說明
- (減號)	實線 (預設)	:	由點連成的線段
--	虛線	none	不繪出線段
-.	虛線和點連成的線段		

表 4.1.5 plot 函數的控制碼（三），控制線條顏色

線條顏色	說明	線條顏色	說明
g	綠色 (green)	w	白色 (white)
m	紫色 (magenta)	r	紅色 (red)
b	藍色 (blue) (預設)	k	黑色 (black)
c	青藍色 (cyan)	y	黃色 (yellow)

○ 簡單的範例：

```
>> x=linspace(1,8,36);
```

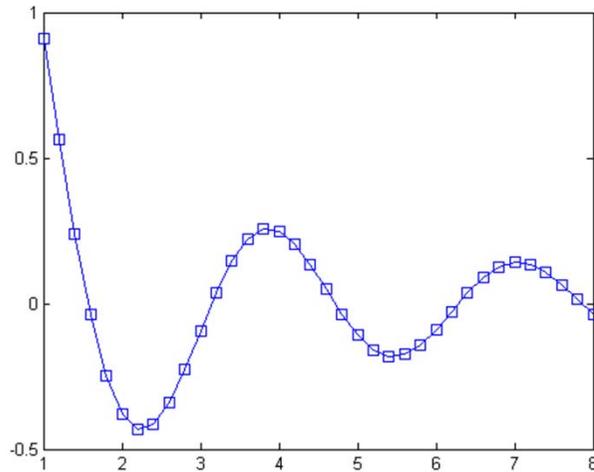
自1到8取36點
形成36行的向量陣列

```
>> y1=sin(2*x)./x;
```

每一點x值*2取sin函數後除以原值
即向量元素相除

```
>> plot(x,y1,'-sb')
```

若沒有.則表示向量投影長度
即 $a/b = \text{dot}(a,b)/\text{dot}(b,b)$





繪圖區域的控制

更改繪圖的範圍與顯示方式

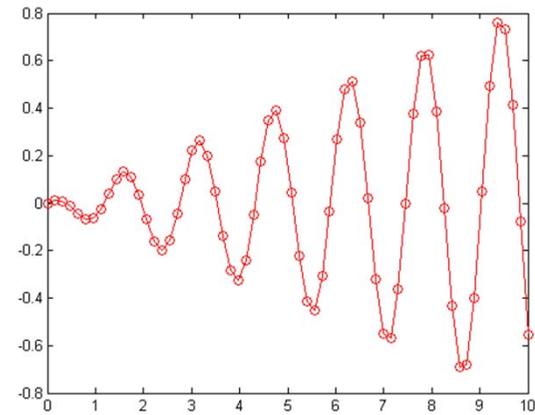
- 如果想自行設定函數圖形顯示的範圍時，則可利用`axis`函數：

表 4.2.1 設定繪圖的範圍

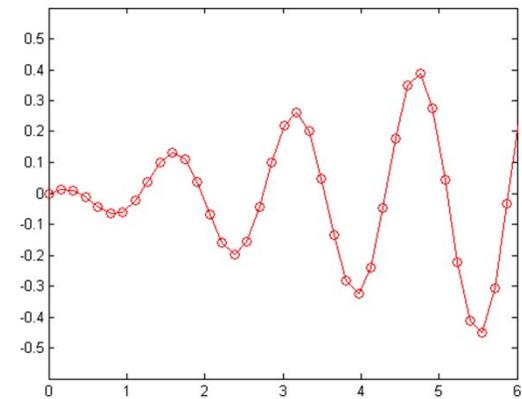
函數	說明
<code>axis([xmin, xmax, ymin, ymax])</code>	指定繪圖的範圍， x 方向從 $xmin$ 到 $xmax$ ， y 方向從 $ymin$ 到 $ymax$



```
>> x=linspace(0,10,64);  
>> y=x.*cos(4*x)./12;  
>> plot(x,y,'-ro')
```



```
>> axis([0,6,-0.6,0.6])
```

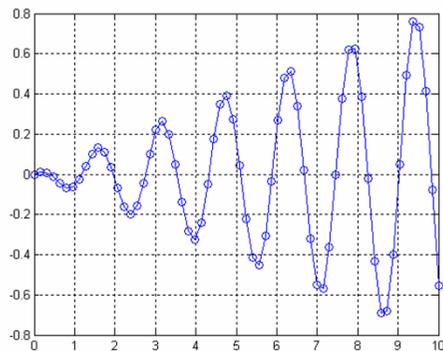


- 利用 `box` 和 `grid` 指令可設定設定格線與外框：

表 4.2.2 設定是否顯示圖形的格線與外框

指令	說明
<code>grid</code>	設定是否顯示格線，設定 <code>on</code> 為顯示，設定 <code>off</code> 則不顯示
<code>box</code>	設定是否顯示圖形的的外框，設定 <code>on</code> 顯示，設定 <code>off</code> 不顯示

```
>> grid on
```





修改x與y軸的顯示比例

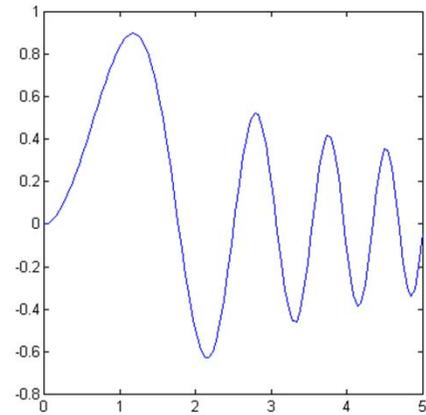
- 設定座標軸顯示的比例所用的指令：

表 4.2.3 設定座標軸顯示的比例

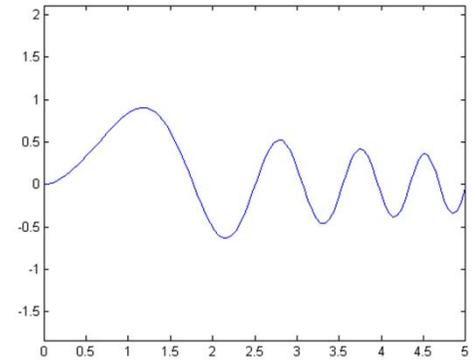
指令	說明
<code>axis normal</code>	使用 Matlab 預設的寬高比，且拉動視窗即可調整其比例
<code>axis square</code>	圖形輸出的寬與高比例為 1:1
<code>axis equal</code>	圖形座標軸的比例為 1:1
<code>axis tight</code>	圖形的繪圖區域緊貼著視窗



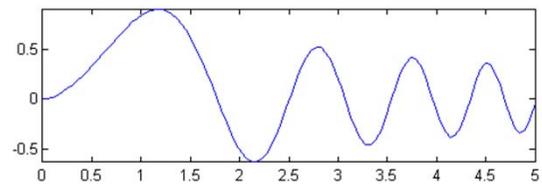
>> axis square



>> axis equal



>> axis equal tight



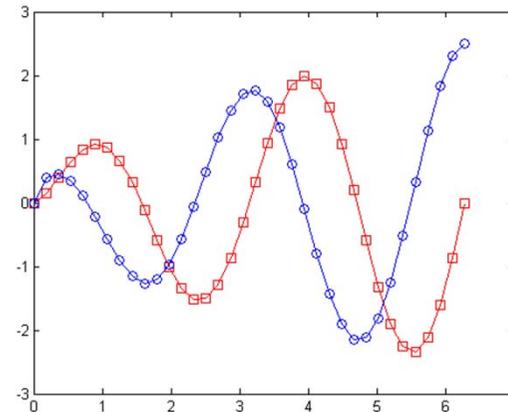
於已存在的圖中加入新圖

- 利用 `hold on` 可將新繪的圖形附加於原有圖形之上：

表 4.2.4 設定圖形產生的方式

指令	說明
<code>hold</code>	設定 <code>hold</code> 為 <code>on</code> 時，則新產生的圖形會疊加在原有圖形的上面，若是設定 <code>off</code> ，則原有的圖形會被新產生的圖形覆蓋掉。

```
>> plot(x,y1,'-rs')  
>> hold on  
>> plot(x,y2,'-bo')
```





建立一個新的繪圖視窗來繪圖

- `figure`指令可另起一個新的視窗來容納新的圖形：

表 4.2.5 設定圖形產生的方式

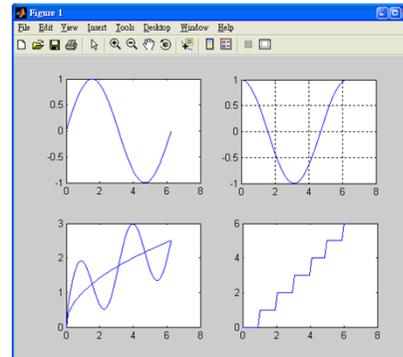
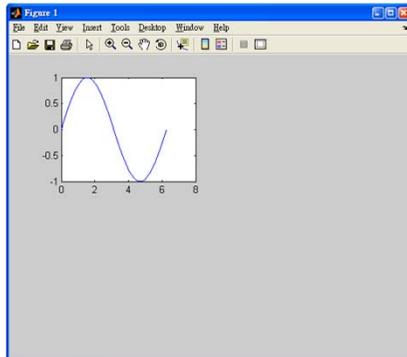
指令	說明
<code>figure</code>	建立一個新的繪圖視窗，視窗的標題為 Matlab 自動設定
<code>figure(<i>n</i>)</code>	建立一個新的繪圖視窗，視窗的標題為 Figure <i>n</i> 。若 Figure <i>n</i> 為已經存在的視窗，則 <code>figure(<i>n</i>)</code> 會把此視窗變成作用中視窗

將數張圖合併成一張大圖

- 利用subplot可在一個繪圖視窗內呈現數張小圖：

表 4.2.6 subplot 函數的用法

函數	說明
<code>subplot(m, n, p)</code>	把繪圖視窗分成 $m \times n$ 個區域，並在第 p 個位置建立一個子繪圖區。位置 p 的計算方式是由左而右，由上而下來排列
<code>subplot(m, n, p, 'replace')</code>	於第 p 個位置建立一個子繪圖區，若此繪圖區內已有其它圖形存在，則新繪的圖會取代掉原有的圖





於圖形內加入文字

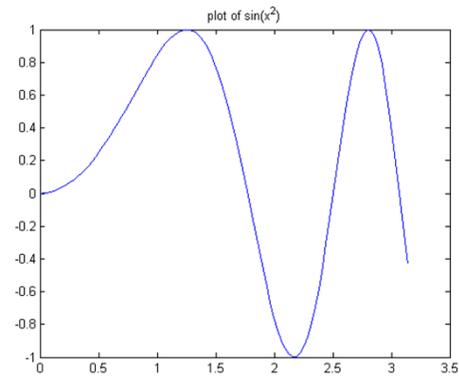
- 下表面的函數可設定圖形的標題文字，以及每一個繪圖軸的解說文字：

表 4.3.1 於圖形內加入文字

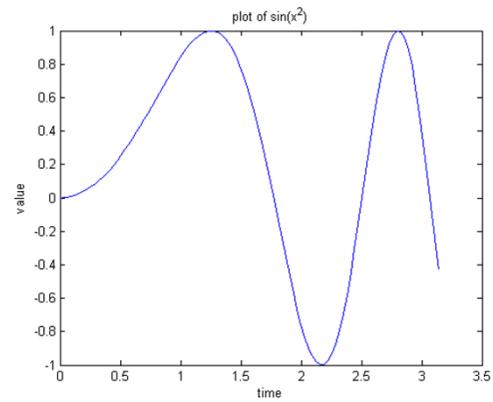
函 數	說 明
<code>title('text')</code>	設定圖形的標題文字為 <i>text</i>
<code>xlabel('text')</code>	設定 <i>x</i> 軸的解說文字為 <i>text</i>
<code>ylabel('text')</code>	設定 <i>y</i> 軸的解說文字為 <i>text</i>
<code>zlabel('text')</code>	設定 <i>z</i> 軸的解說文字為 <i>text</i>



```
>> title('plot of sin(x^2)')
```



```
>> xlabel('time');ylabel('value');
```



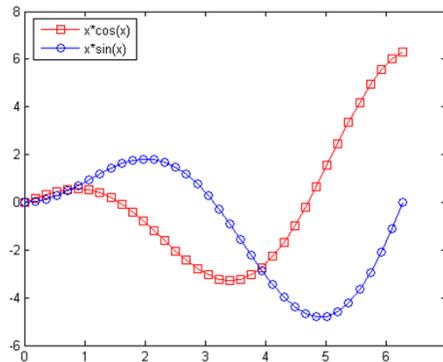
- 
-
- 下面的函數可在圖形內加入註解：

表 4.3.2 加入圖形的註解

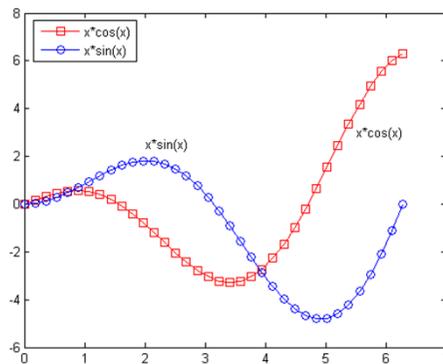
函 數	說 明
<code>legend(str₁, str₂, ...)</code>	設定圖例標記的字串
<code>legend(str₁, str₂, ..., pos)</code>	設定圖例標記的位置，1 代表將圖例放在右上角，2 是左上角，3 是左下角，4 則是放在右下角
<code>legend off</code>	清除圖例標記
<code>text(x, y, 'text')</code>	在圖形中位置為(x, y)之處加入註解文字
<code>gtext('text')</code>	利用滑鼠來設定文字輸入的位置



```
>> legend('x*cos(x)', 'x*sin(x)', 2)
```



```
>> text(2, 2.5, 'x*sin(x)'); text(5.5, 3, 'x*cos(x)')
```





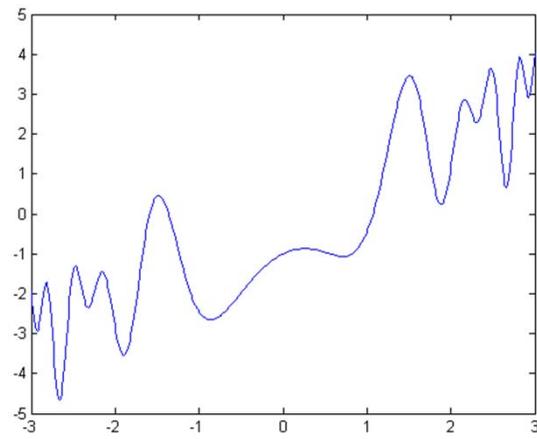
更簡潔的繪圖函數

- `fplot`函數只要給予一個函數字串即可繪圖
- `fplot`可依據圖形陡峭的程度，自動調整樣點數的多寡以繪出平滑的曲線

表 4.4.1 繪圖函數 `fplot` 的用法

函數	說明
<code>fplot('f_str', [xmin, xmax])</code>	繪出函數 f_str 的圖形， x 軸的範圍取 $xmin$ 到 $xmax$
<code>fplot('f_str', [xmin, xmax, ymin, ymax])</code>	繪出函數 f_str 的圖形， x 軸的範圍取 $xmin$ 到 $xmax$ ， y 軸的範圍取 $ymin$ 到 $ymax$

```
>> fplot('x-cos(x^3)-sin(2*x^2)',[-3,3])
```



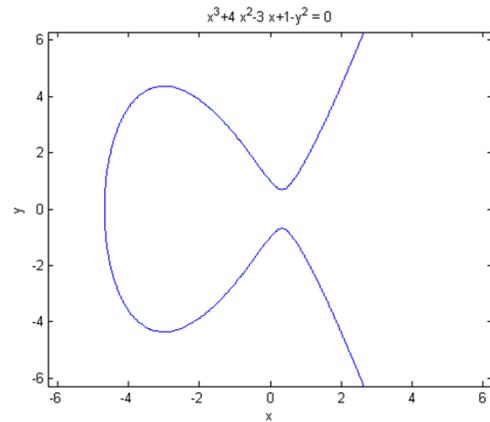
- 
-
- `ezplot`可繪出隱函數圖，以及參數繪圖等：

表 4.4.2 繪圖函數 `ezplot` 的用法

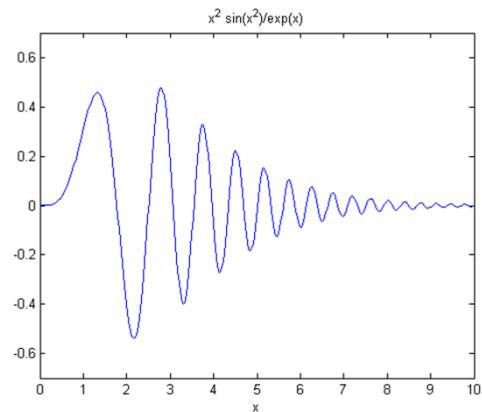
函 數	說 明
<code>ezplot('f_str', [xmin, xmax])</code>	繪出函數 f_str 的圖形，繪圖範圍在 x 與 y 方向均取 $xmin$ 到 $xmax$
<code>ezplot('f_str', [xmin, xmax, ymin, ymax])</code>	繪出函數 f_str 的圖形，繪圖範圍在 x 方向取 $xmin$ 到 $xmax$ 在 y 方向均取 $ymin$ 到 $ymax$
<code>ezplot('fx', 'fy', [tmin, tmax])</code>	參數繪圖，繪出 $(fx(t), fy(t))$ ， t 從取 $tmin$ 到 $tmax$ 的參數圖



```
>> ezplot('x^3+4*x^2-3*x+1-y^2')
```

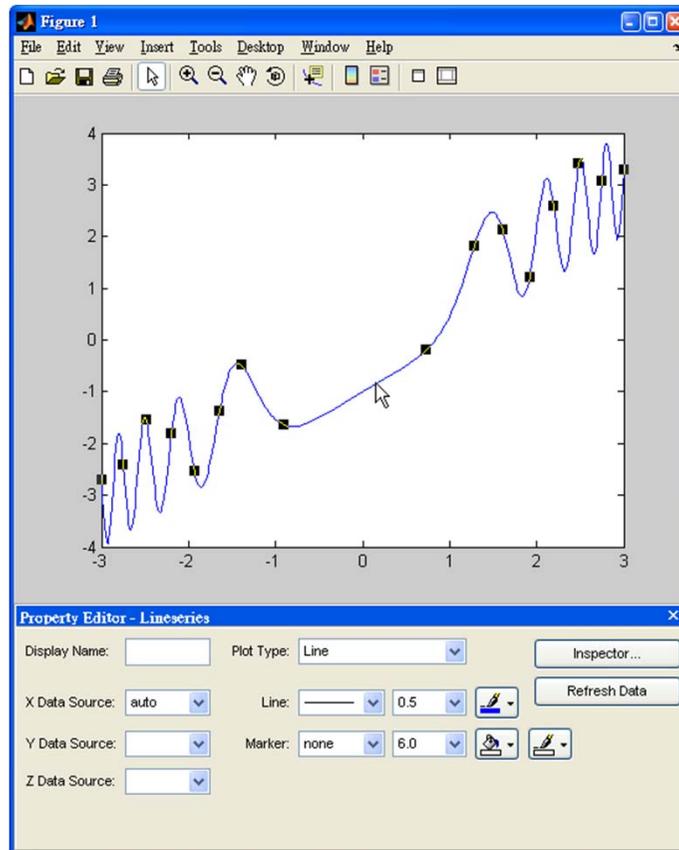


```
>> ezplot('x^2*sin(x^2)/exp(x)', [0, 10, -0.7, 0.7])
```

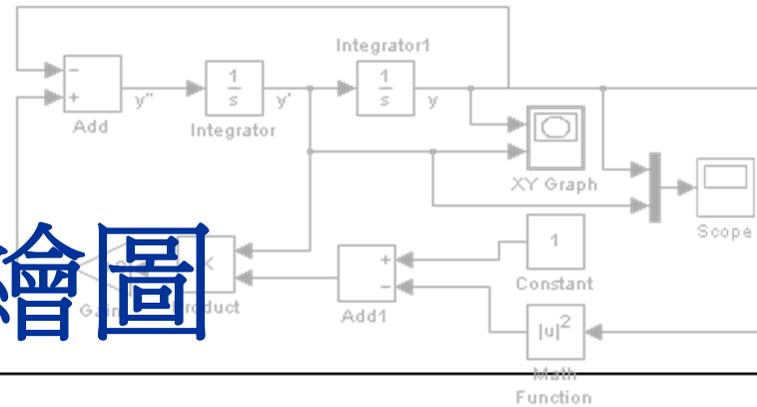


利用Property Editor來編修圖形

- 利用Property Editor對話方塊可編修圖形的性質：



三維空間繪圖

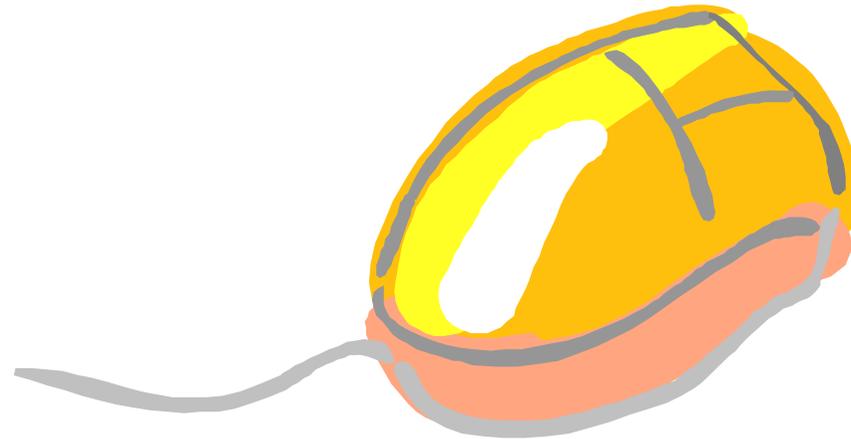


學習三維繪圖的基本技巧

學習peaks函數的用法

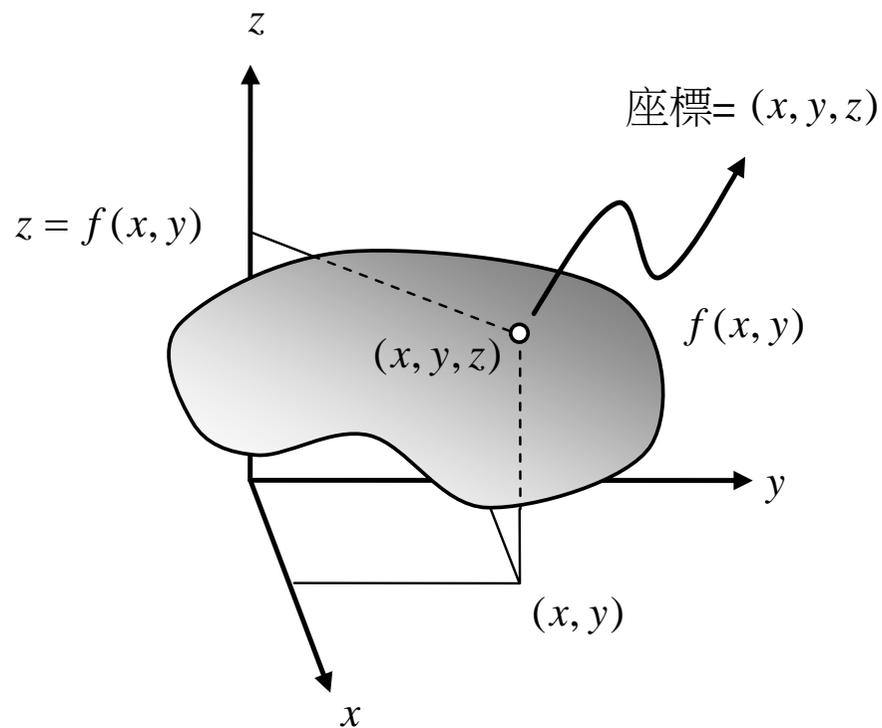
學習二維與三維等高線圖的繪製

學習三維圖形的編修



基本三維繪圖

- 對於函數而言，每給一組 $f(x,y)$ ，便能求得其相對應的高度 z ，如下圖所示：





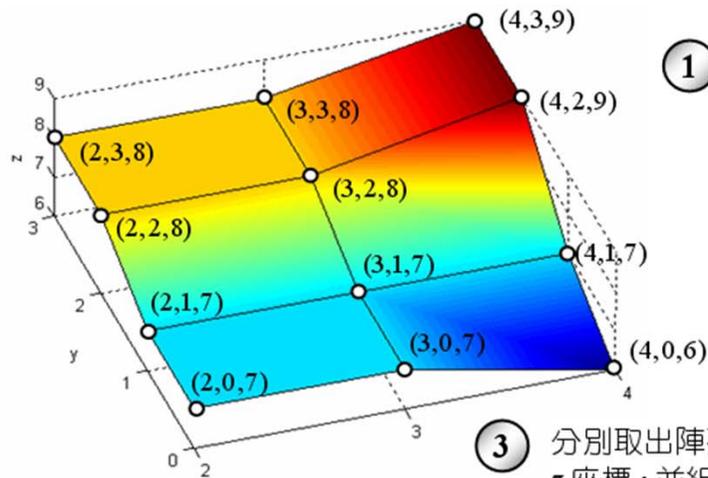
繪製三維的網格圖

- 利用mesh函數可繪製三維的網格圖：

表 5.1.1 mesh 函數的使用

函 數	說 明
<code>mesh(x,y,z)</code>	繪出三維的網格圖
<code>mesh(z)</code>	繪出 x 座標從 1 到 m ， y 座標從 1 到 n 的三維的網格圖

Mesh 函數的用法



① 從圖形上方觀測，並標上資料點的座標

② 將資料點的座標取出，並依序組成一個陣列 A

$$A = \begin{bmatrix} (2, 3, 8) & (3, 3, 8) & (4, 3, 9) \\ (2, 2, 8) & (3, 2, 8) & (4, 2, 9) \\ (2, 1, 7) & (3, 1, 7) & (4, 1, 7) \\ (2, 0, 7) & (3, 0, 7) & (4, 0, 6) \end{bmatrix}$$

③ 分別取出陣列 A 的 x 、 y 與 z 座標，並組成三個陣列

$$xx = \begin{bmatrix} 2 & 3 & 4 \\ 2 & 3 & 4 \\ 2 & 3 & 4 \\ 2 & 3 & 4 \end{bmatrix}$$

將橫列反向排序

$$\begin{bmatrix} 3 & 3 & 3 \\ 2 & 2 & 2 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$yy =$

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \end{bmatrix}$$

將橫列反向排序

$$\begin{bmatrix} 8 & 8 & 9 \\ 8 & 8 & 9 \\ 7 & 7 & 7 \\ 7 & 7 & 6 \end{bmatrix}$$

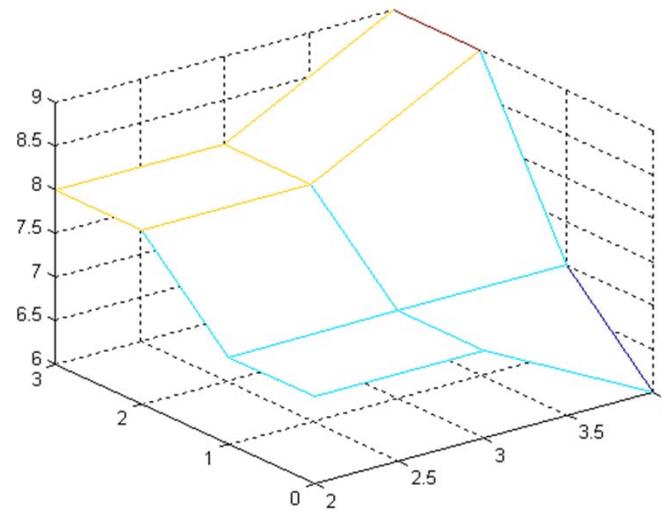
$zz =$

$$\begin{bmatrix} 7 & 7 & 6 \\ 7 & 7 & 7 \\ 8 & 8 & 9 \\ 8 & 8 & 9 \end{bmatrix}$$

④ 利用 `mesh` 指令來繪製三維的網格圖

`mesh (xx, yy, zz)`

```
>> 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=[7 7 6;7 7 7;8 8 9;8 8 9]  
>> mesh(xx,yy,zz)
```

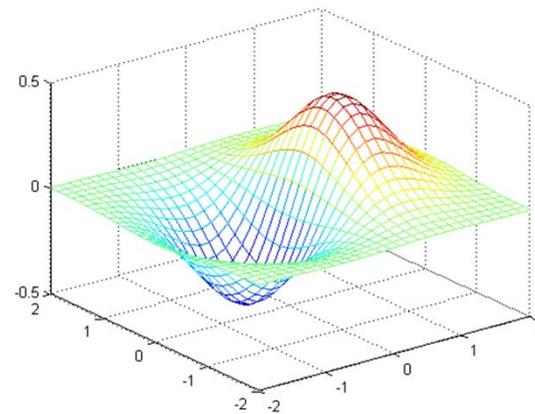


-
- Matlab提供了meshgrid函數，可以建立 xx 與 yy 矩陣：

表 5.1.2 meshgrid 函數的使用

函 數	說 明
meshgrid(vx,vy)	建構出兩個二維矩陣 xx 與 yy ，以供三維繪圖所需

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

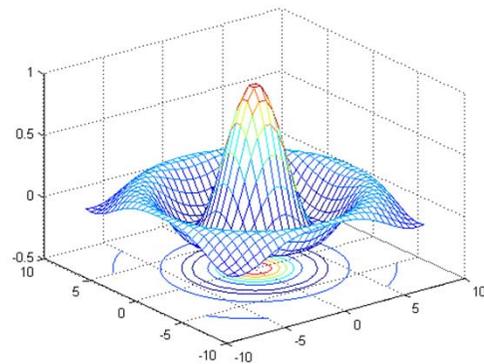


○ 下面是meshc函數與waterfall函數用法：

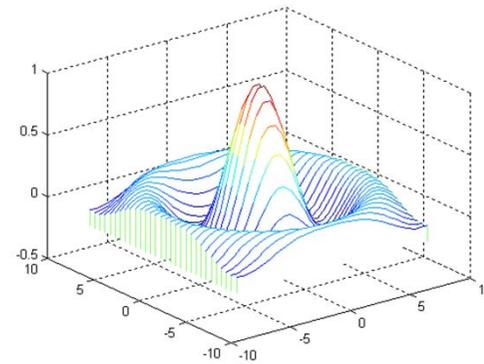
表 5.1.3 meshc 與 waterfall 函數的使用

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

>> meshc(xx,yy,zz)



>> waterfall(xx,yy,zz)



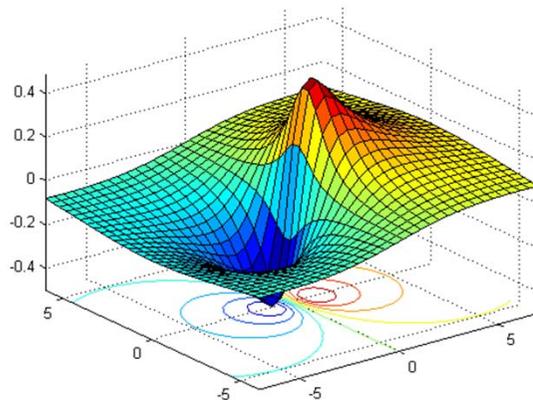
繪製三維的曲面圖

- 想要對網格面上色，可利用surf或其它相關的函數：

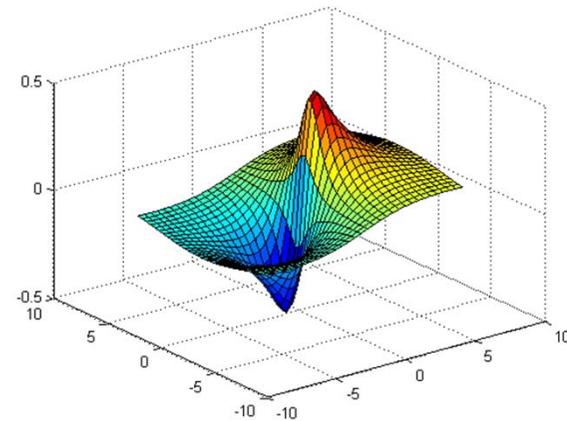
表 5.1.4 surf 與 surfc 函數的使用

函數	說明
<code>surf(xx,yy,zz)</code>	繪出三維的曲面圖
<code>surfc(xx,yy,zz)</code>	同 <code>surf</code> ，但在圖形下方會顯示出函數圖形的等高線圖

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



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





簡易的三維繪圖函數

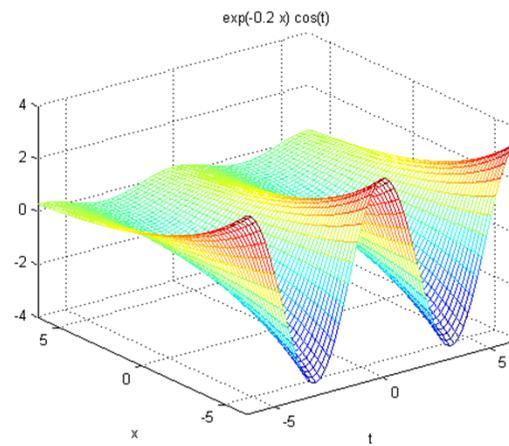
- `ezmesh`與`ezsurf`函數可以快速的繪出三維的圖形：

表 5.2.1 簡易三維繪圖函數的使用

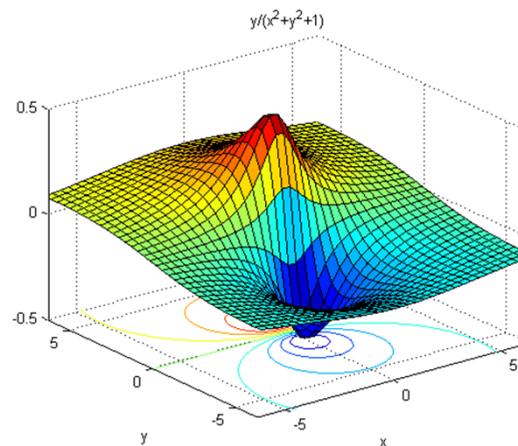
函 數	說 明
<code>ezmesh(f, [xmin, xmax, ymin, ymax])</code>	根據函數 f 以 60×60 個網格數繪出 f 的三維圖形
<code>ezmeshc(f, [xmin, xmax, ymin, ymax])</code>	同 <code>ezmesh</code> ，但在圖形下方會顯示出圖形的等高線
<code>ezsurf(f, [xmin, xmax, ymin, ymax])</code>	同 <code>ezmesh</code> ，但是網格面會上色
<code>ezsurfc(f, [xmin, xmax, ymin, ymax])</code>	同 <code>ezsurf</code> ，但在圖形下方會顯示出圖形的等高線



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



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



內建的三維圖形展示函數—peaks

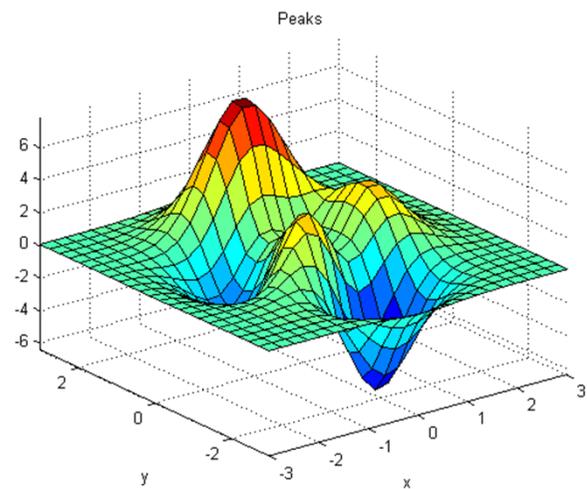
- peaks所描述的數學函數，其定義式為

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

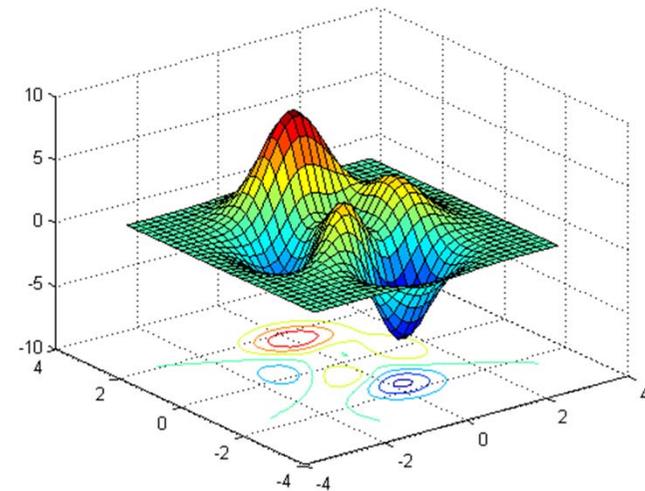
表 5.3.1 使用 peaks 函數

函數	說明
peaks	以 49×49 個資料點繪製數學函數 peaks，範圍 x 與 y 方向同為 $-3 \sim 3$
peaks(n)	同 peaks，但以 $n \times n$ 個資料點來繪圖
zz=peaks	計算 49×49 個數學函數 peaks 的值
zz=peaks(n)	以 $n \times n$ 個資料點計算數學函數 peaks 的值
[xx,yy,zz]=peaks(n)	以 $n \times n$ 個資料點計算數學函數 peaks 的值

```
>> peaks(24);  
>> surfc(xx,yy,zz);
```



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



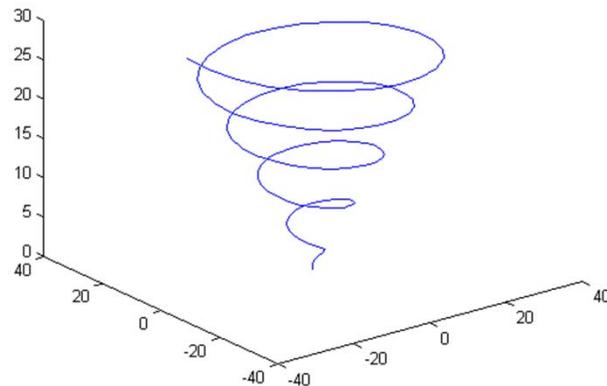
空間曲線繪圖

- `plot3` 函數可用來繪製空間的曲線：

表 5.4.1 空間曲線繪圖函數

函 數	說 明
<code>plot3(x,y,z)</code>	以向量 x, y 與 z 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

```
>> plot3(t.*sin(t),t.*cos(t),t);
```





等高線繪圖

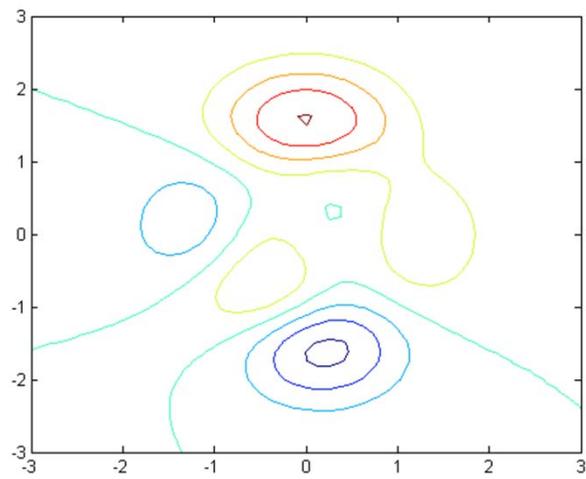
二維等高線圖

- Matlab的contour函數可用來繪製二維的等高線圖：

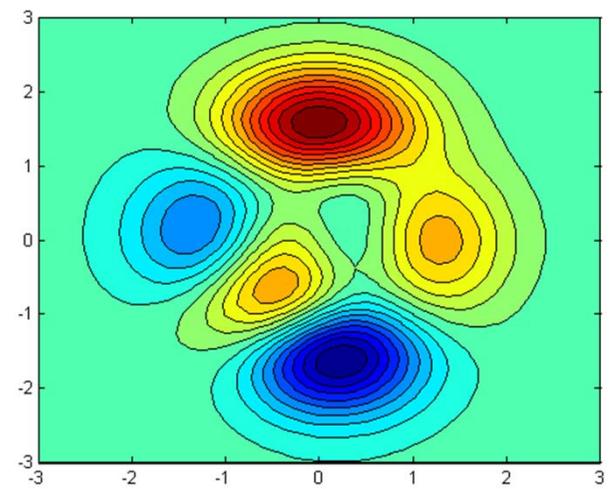
表 5.5.1 二維等高線繪圖函數

函 數	說 明
<code>contour(xx,yy,zz,n)</code>	分別以矩陣 xx 、 yy 與 zz 繪出 n 條等高線
<code>contour(zz,n)</code>	同上，但 x 方向是從 1 到 m ， y 方向是從 1 到 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> 函數，但會以顏色填滿 (fill) 等高線圖

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



```
>> contourf(xx,yy,zz,20)
```





○ 要標註等高線的值：

把繪出的等高線圖設給某一個變數

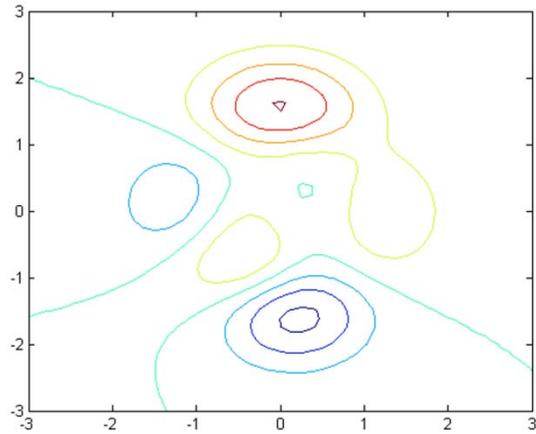
把這個變數傳遞給`clabel`函數

表 5.5.2 將等高線加入高度標記的函數

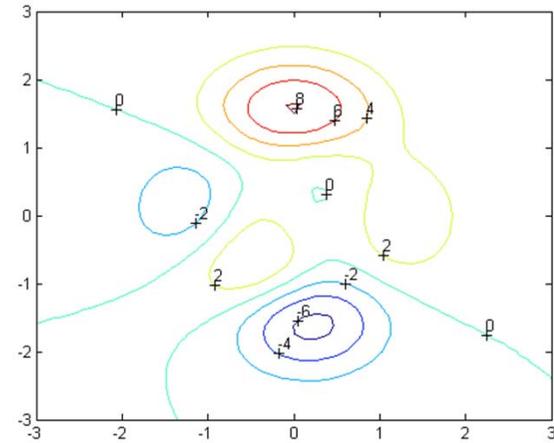
函數	說明
<code>clabel(cmat)</code>	在等高線圖內加上高度的標記
<code>clabel(cmat, [z₁, z₂, z₃, ...])</code>	在高度為 $[z_1, z_2, z_3, \dots]$ 的等高線上加上高度標記
<code>clabel(cmat, 'manual')</code>	利用滑鼠標註等高線的數值



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



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



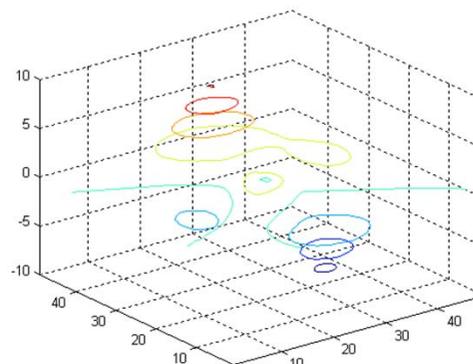
三維的等高線圖

- `contour3` 函數可繪製三維的等高線圖：

表 5.5.3 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 <code>xx</code> 、 <code>yy</code> 與 <code>zz</code> 繪出 n 條三維的等高線
<code>contour3(zz,n)</code>	同上，但 x 方向從 1 到 m ， y 方向從 1 到 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);
```





編修三維繪圖

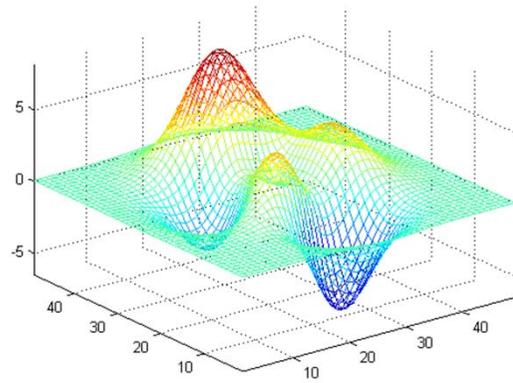
三維圖形的基本編修

- 下表列出了三維圖形常用的編修指令：

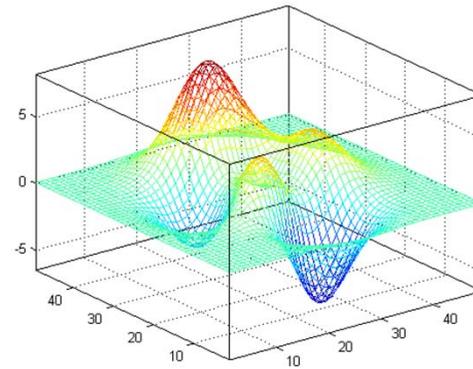
表 5.6.1 三維繪圖的基本編修指令

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

`>> hidden off;`



`>> box on;`

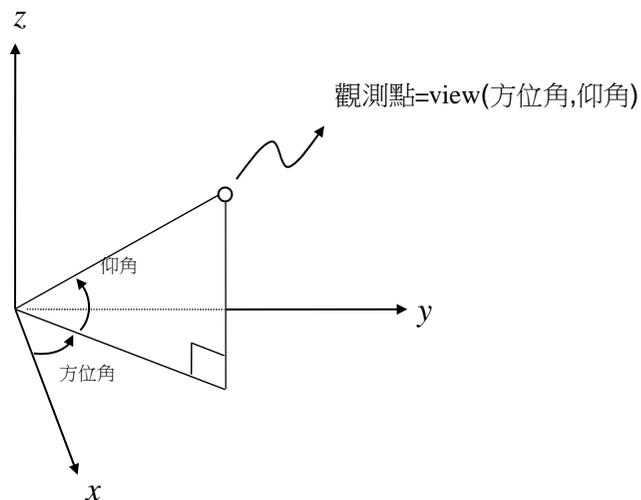


改變三維圖形的視角

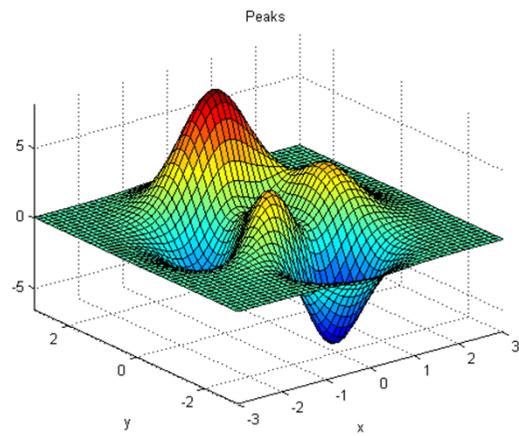
- 如果想更改圖形的觀測角度，可用利用view函數：

表 5.6.2 改變三維圖形的視角

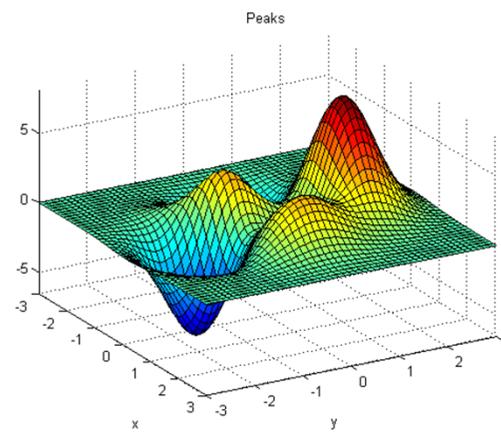
函 數	說 明
<code>view(az,el)</code>	設定圖形的視角，單位為度
<code>[az,el]=view</code>	傳回目前所使用的視角



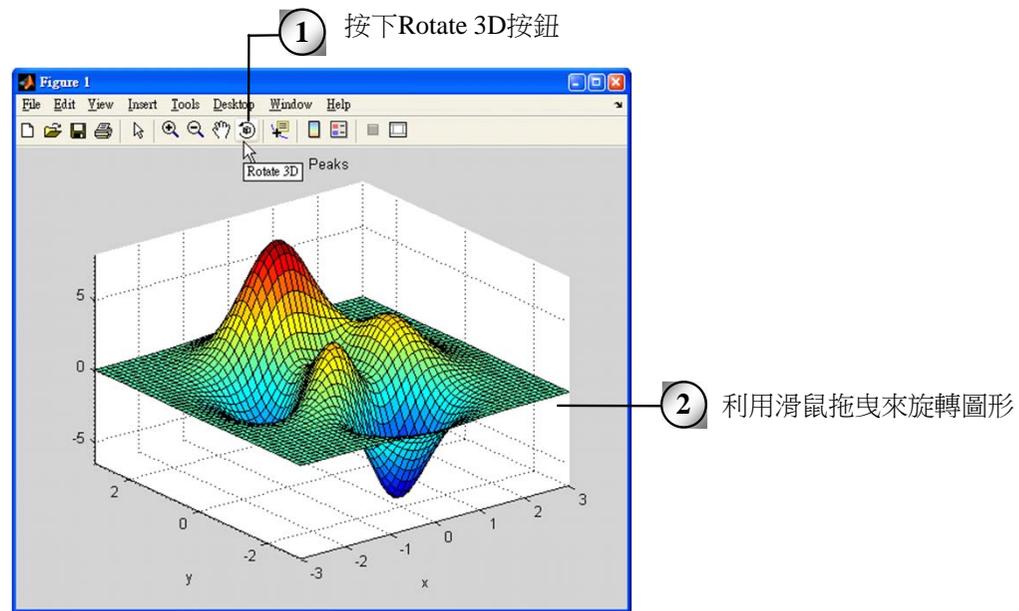
>> peaks;



>> view(60,30);

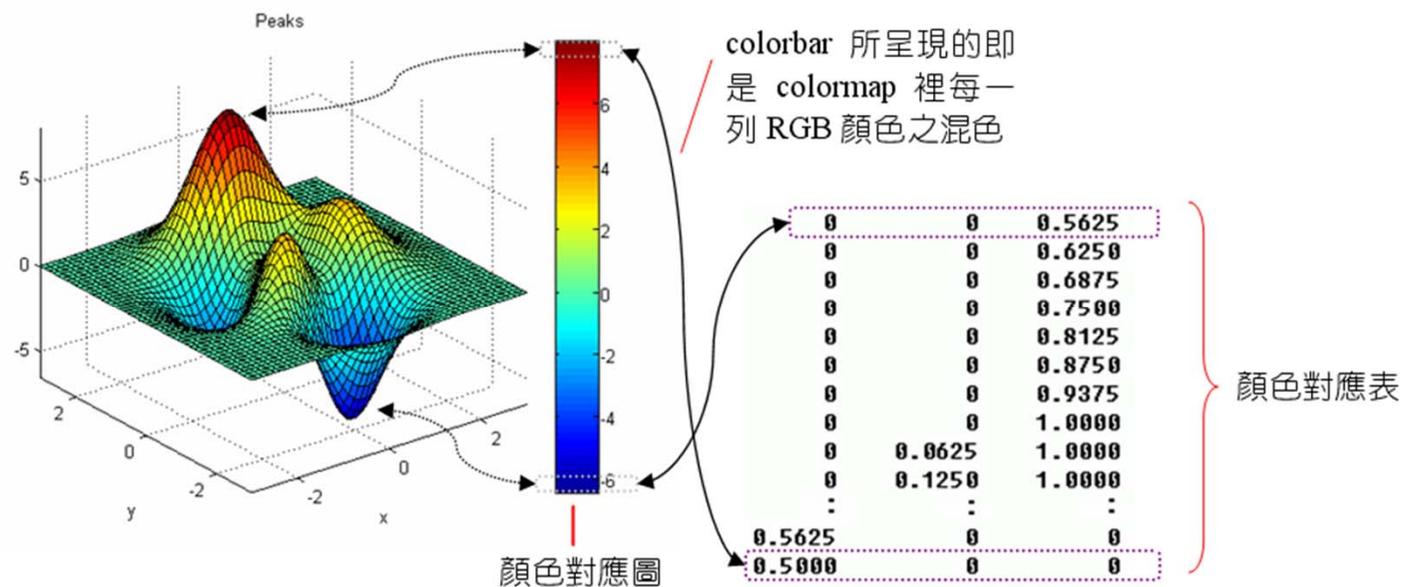


- 按下工具列上的Rotate 3D鈕，可利用滑鼠旋轉所繪製的圖形



修改三維圖形的曲面顏色

- Matlab是利用color map，依所繪製之函數值的大小來對曲面上色。



- 
-
- `colormap`可限定三維的圖形使用特定的顏色對應表
 - 下表列出了`colormap`與`colorbar`函數的用法：

表 5.6.4 `colormap` 函數的使用

函 數	說 明
<code>colormap(<i>map</i>)</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><i>map</i>=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

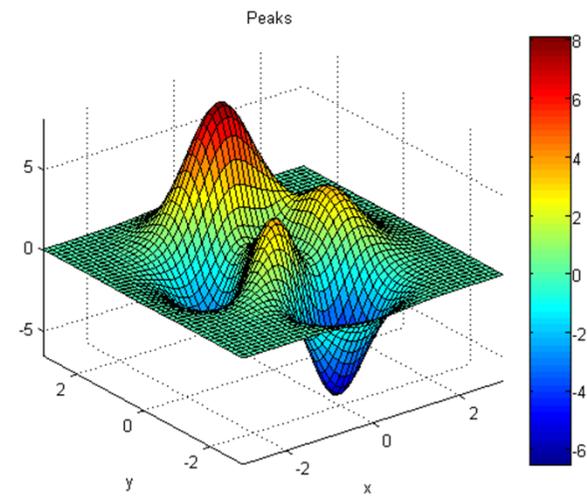
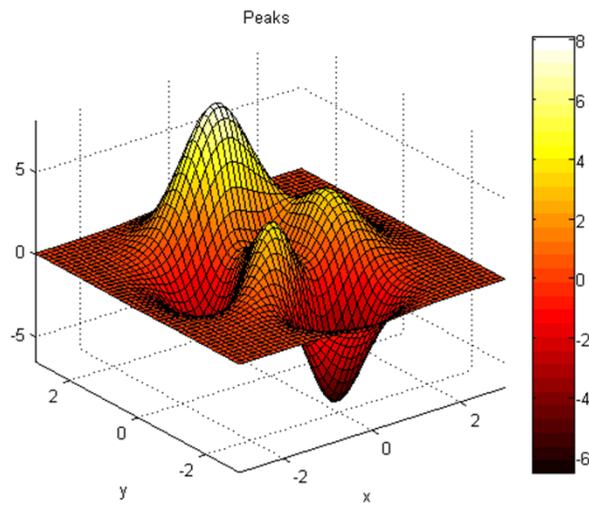


○ 下表列出了Matlab常用來建立顏色對應表的函數

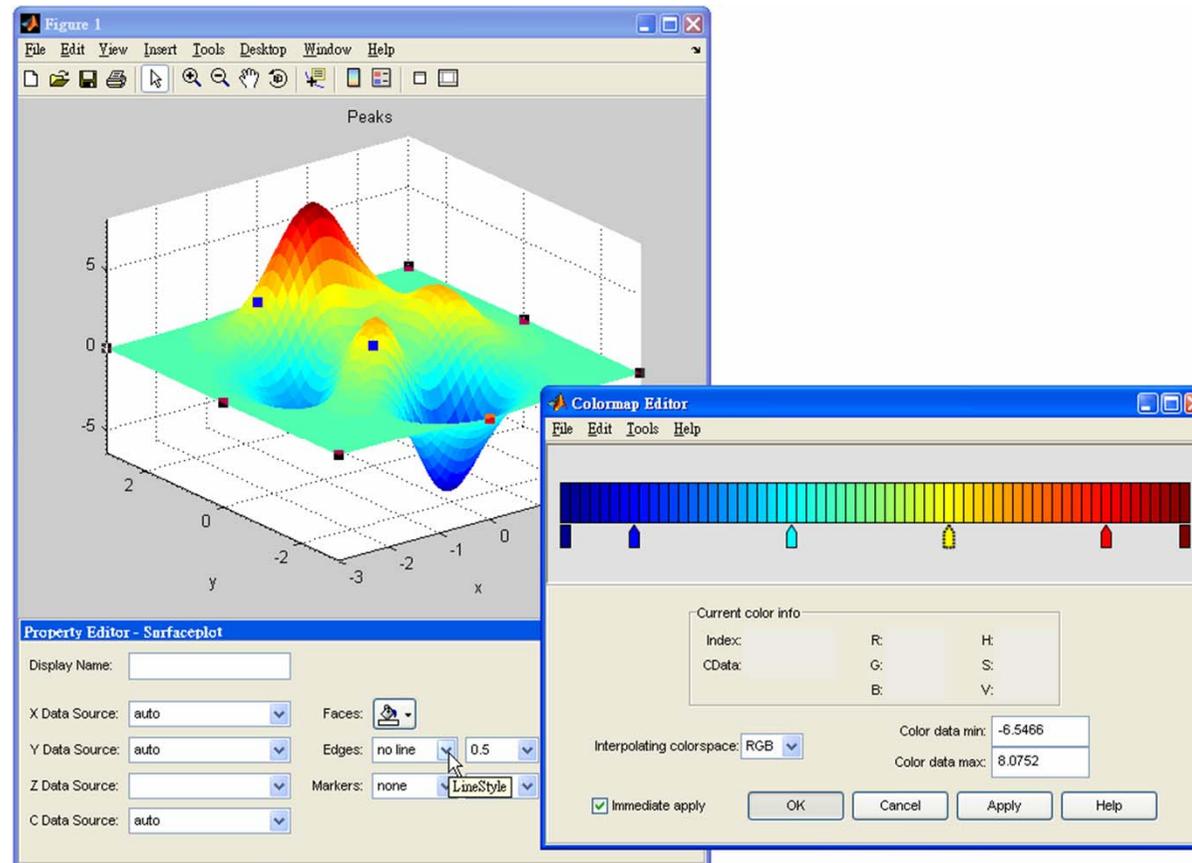
表 5.6.5 產生顏色對應表的函數

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

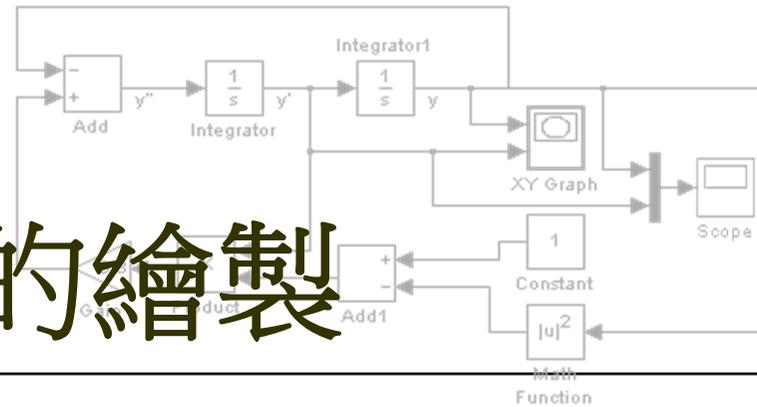
```
>> colormap(hot(32));colorbar; >> colormap('default');colorbar;
```



利用Property Editor視窗修改圖形



特殊圖形的繪製



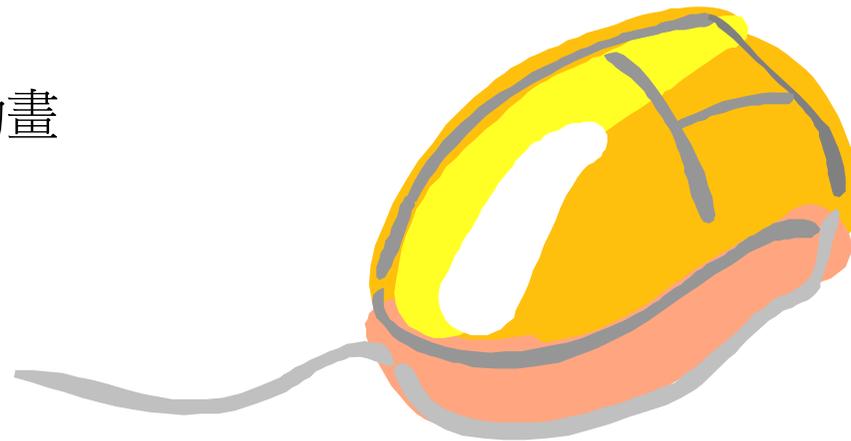
學習極座標繪圖與對數繪圖

學習雙 y 軸繪圖

學習向量場繪圖

學習統計繪圖

在Matlab的環境裡製作動畫





常用的二維繪圖函數

極座標繪圖

- 極座標函數可以寫成

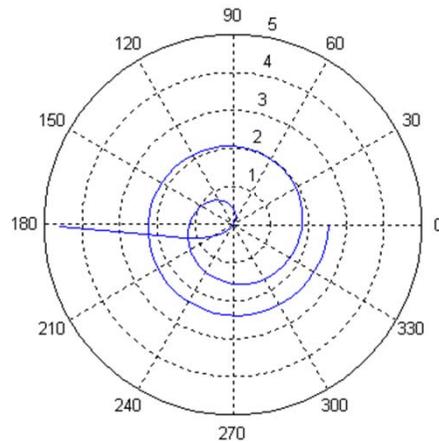
$$r = f(\theta)$$

- Matlab 的 `polar` 函數來繪製極座標圖。

表 6.1.1 `polar` 函數的使用

函數	說明
<code>polar(theta, r)</code>	根據角度向量 <i>theta</i> ，以及距原點的長度 <i>r</i> 繪製極座標圖
<code>polar(theta, r, 'str')</code>	依據格式字串 <i>str</i> 所指定的格式繪製極座標圖

```
>> t=linspace(0.01,4*pi,100);  
>> r=log(t);  
>> polar(t,r)
```



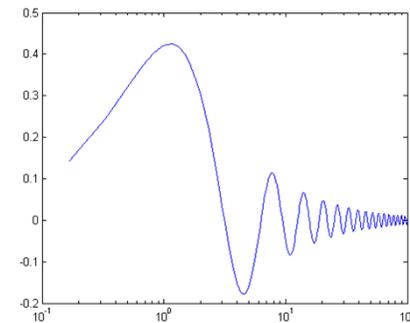
對數繪圖

- 下表所列的函數可繪製對數座標圖形：

表 6.1.2 對數繪圖函數的使用

函數	說明
<code>semilogx(x,y)</code>	x 軸為對數座標，繪出 $x-y$ 的對數圖
<code>semilogy(x,y)</code>	y 軸為對數座標，繪出 $x-y$ 的對數圖
<code>loglog(x,y)</code>	x 軸與 y 軸皆為對數座標，繪出 $x-y$ 的對數圖

```
>> x=linspace(0,100,600);  
>> semilogx(x,sin(x)./(x+1))
```



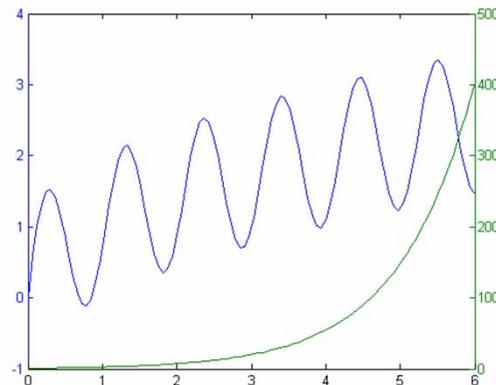
雙 y 軸繪圖

- 利用雙 y 軸繪圖可繪製兩個函數於同一張圖：

表 6.1.3 plotyy 函數的使用

函數	說明
<code>plotyy(x1,y1,x2,y2)</code>	以圖形左邊的刻度當成 x_1 - y_1 資料點的 y 軸，以圖形右邊的刻度當成 x_2 - y_2 資料點的 y 軸，繪出雙 y 軸圖

```
>> x=linspace(0,6,50);  
>> plotyy(x,sqrt(x)+sin(6*x),x,exp(x))
```





向量場與法向量繪圖

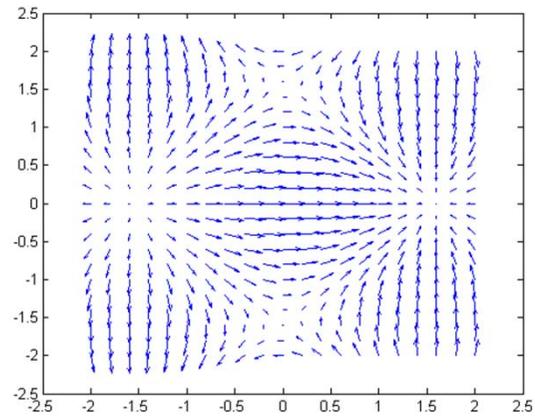
梯度向量場的繪製

- 要繪出梯度向量場
以 `gradient` 函數計算
再以 `quiver` 函數繪出圖形

表 6.2.1 `gradient` 與 `quiver` 函數的語法

函 數	說 明
<code>[fx, fy]=gradient(zz)</code>	依矩陣 <code>zz</code> 計算出每一個資料點的梯度
<code>[fx, fy]=gradient(zz, dx, dy)</code>	同上，但 x 軸方向間距是 dx ， y 軸方向間距是 dy
<code>quiver(xx, yy, fx, fy)</code>	在座標為 xx 與 yy 的點上繪出一個箭號

```
>> [xx,yy]=meshgrid(-2:0.2:2,-2:0.2:2);  
>> zz=sin(xx).*cos(yy);  
>> [u,v]=gradient(zz);  
>> quiver(xx,yy,u,v)
```



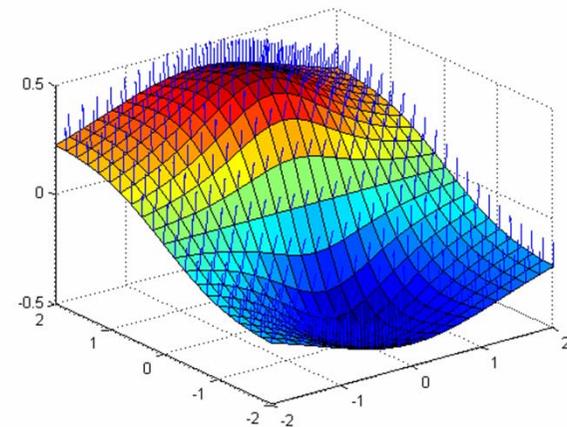
三維法向量的繪圖

- 要繪製曲面的法向量，可利用 `surfnorm` 與 `quiver3` 函數：

表 6.2.3 `quiver3` 的用法與三維的法向量繪圖

函數	說明
<code>surfnorm(xx,yy,zz)</code>	利用 <code>xx,yy</code> 與 <code>zz</code> 所描述的曲面計算其法向量
<code>quiver3(xx,yy,zz,fx,fy,fz)</code>	同 <code>quiver</code> ，但是繪出三維的向量場
<code>quiver3(fx,fy,fz)</code>	同上，但是箭號的間格大小相等

```
>> [u,v,w]=surfnorm(xx,yy,zz);  
>> quiver3(xx,yy,zz,u,v,w,0.4),
```





統計繪圖

長條圖

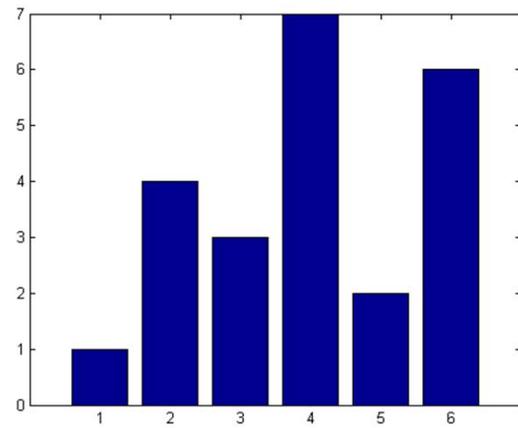
- Matlab提供了bar與bar3函數，可用來繪製二維與三維的長條圖：

表 6.3.1 長條圖繪圖函數

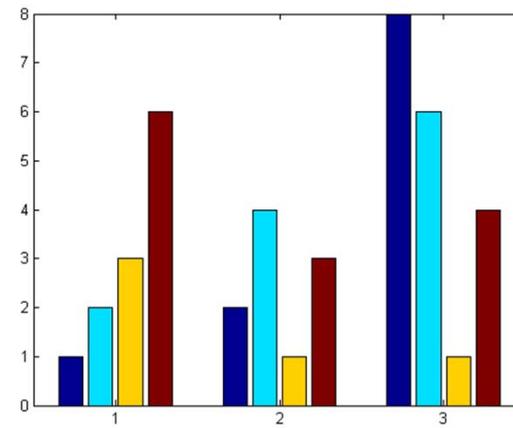
函 數	說 明
<code>bar(y)</code>	依 y 的值來繪製長條圖
<code>bar(x,y)</code>	指定向量 x 的元素值為座標軸的標記來繪圖
<code>bar(x,y,width)</code>	指定長條圖裡長方形的寬度，預設值為 0.8
<code>bar(x,y,'stacked')</code>	將同一群組的長條圖疊加起來繪圖



```
>> bar([1 4 3 7 2 6])
```



```
>> A=[1 2 3 6;2 4 1 3;8 6 1 4]  
>> bar(A)
```

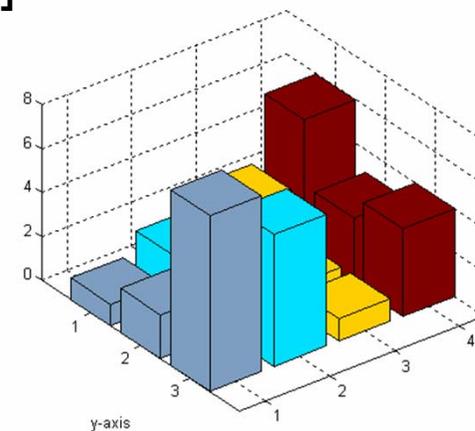


○ **bar3**可用來繪製三維的圖形：

表 6.3.2 立體的長條圖繪圖函數

函數	說明
<code>bar3(zz)</code>	同 <code>bar</code> 函數，但是繪出三維的長條圖
<code>bar3(y,zz)</code>	同上，其中向量 <code>y</code> 可用來指定三維圖中 <code>y</code> 方向的刻度

```
>> A=[1 2 3 6;2 4 1 3;8 6 1 4]
>> bar3(A);ylabel('y-axis')
```



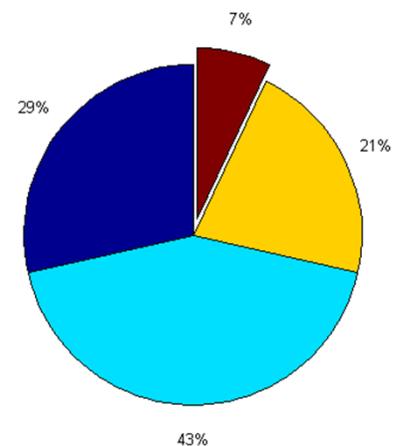
圓形圖

- `pie`與`pie3`函數，可分別繪製二維與三維的圓形圖：

表 6.3.3 圓形圖繪圖

函 數	說 明
<code>pie(x,explode)</code>	依向量 x 繪出圓形圖，並依向量 $explode$ 決定該塊區域是否要和圓形圖分開
<code>pie3(x,explode)</code>	同 <code>pie</code> 指令，但是以三維的方式來呈現

```
>> pie([4 6 3 1],[0 0 0 1])
```





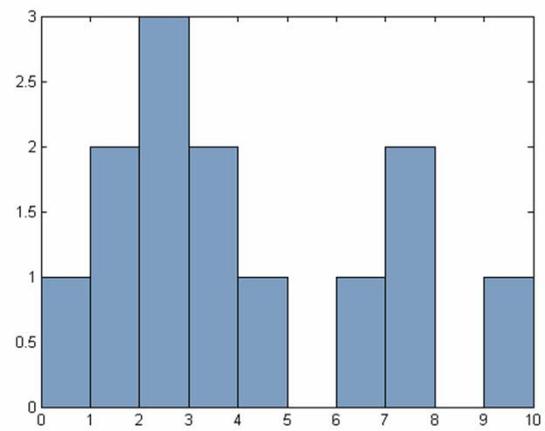
直方圖

- 直方圖（**histogram**）是以組別為橫軸，次數或度為縱軸所繪出的統計圖：

表 6.3.4 直方圖繪圖函數 `hist`

函 數	說 明
<code>v=hist(data)</code>	將向量 <i>data</i> 按數據大小分成 10 個等距的區間，然後將這 10 個區間內元素的個數傳回給向量 <i>v</i>
<code>v=hist(data, n)</code>	同上，但區間數為 <i>n</i>

```
>> data=[0 3 3 4 5 3 7 4 2 8 2 8 10];  
>> hist(data)
```



動畫的製作

- comet函數在繪製動態圖形時，會拖了一條長長的尾巴，因而得名。

表 6.4.1 使用 comet 函數

函數	說明
<code>comet(x,y,p)</code>	繪出彗星軌跡圖，彗星尾巴拖的長度為 $p*\text{length}(y)$
<code>comet3(x,y,z,p)</code>	同上，但繪出三維的彗星軌跡圖

```
>> comet3(sin(t/2).*cos(6*t),sin(t/2).*sin(6*t),t)
```

