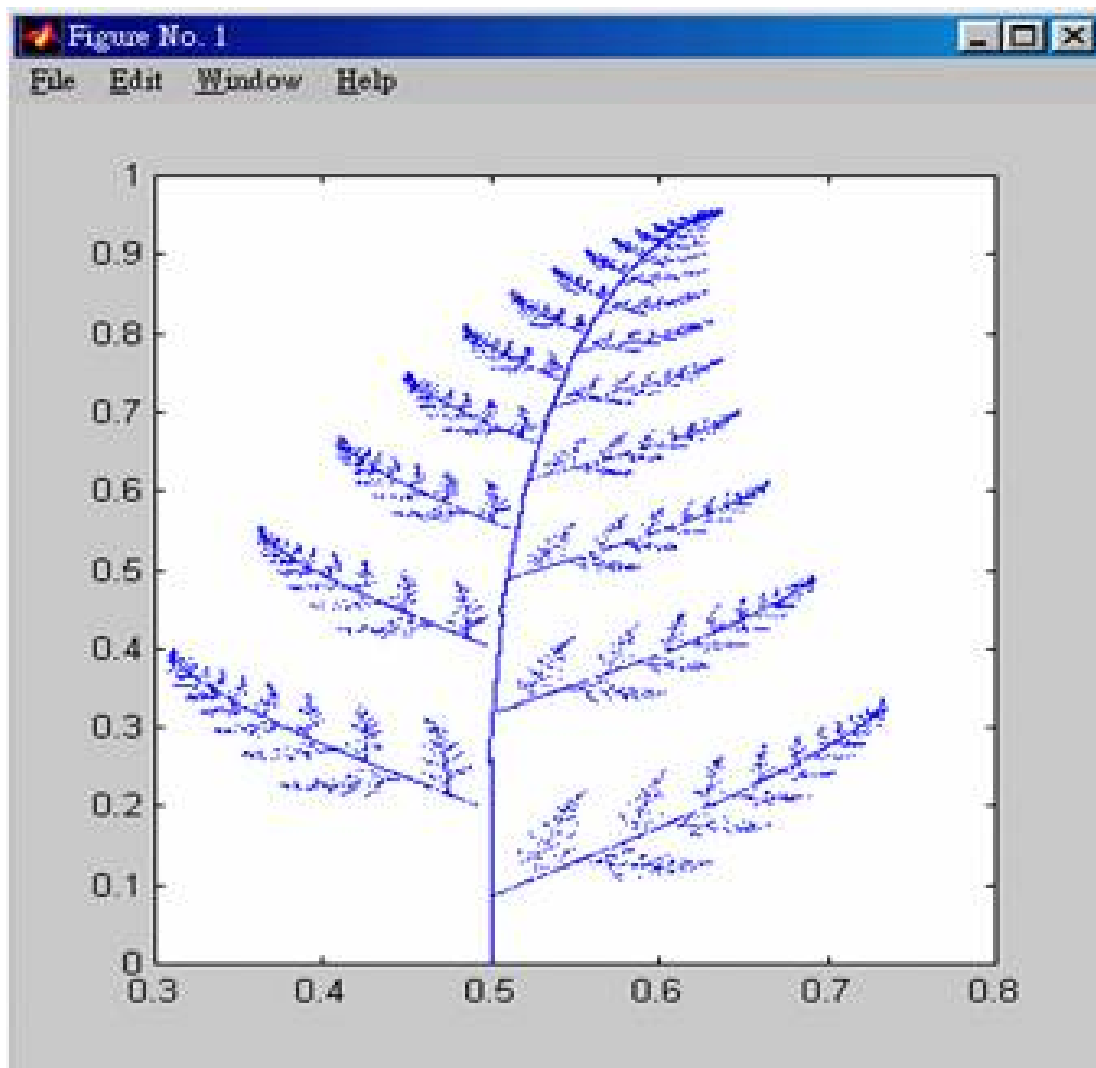


Matlab 在微積分課程之應用



林晶璟 老師

臺北科大 通識中心

檔案已上網 <http://math.ntut.edu.tw>

目 錄

簡 介.....	1
基本使用語法.....	1
一般數學運算.....	2
常用基本函數.....	3
繪圖.....	4
基本繪圖.....	5
二維圖形.....	8
三維圖形.....	9
矩陣基本運算.....	12
常用矩陣運算.....	14
微積分應用.....	16
程式撰寫.....	20
程式流程控制.....	23
應用範例.....	29
基因演算法.....	29

簡介

Matlab 是套應用於科學與工程領域中數值計算、分析與模擬的應用軟體，結合了數值分析、矩陣運算、訊號處理、系統模擬及繪圖等功能，功能強大、操作介面簡易。在大學課程中應用 Matlab 來輔助教學，例如線性代數、控制系統、訊號處理以及微積分。『微積分電腦輔助學習』網站由介紹一般與廣泛的應用方法切入，並以微積分為主要輔助目標。

Matlab 全文為 "matrix laboratory"，原始發展理念是提供完整的矩陣運算指令，將資料元素當成是矩陣來處理，而主要的用途就是作矩陣式的數學運算。隨著日後增加的數學運算需求而增加了系統模擬、數值分析、訊號處理及圖形等功能，也容許使用者建立、運算其他不同的資料型態，滿足更多樣、更深入的運算需求，使 Matlab 應用範圍更為廣大。

本『微積分電腦輔助學習』網站的『Matlab 教學』將針對基本使用方法、一般數學運算、繪圖、矩陣運算、微積分應用等，介紹一些完整的應用範例。使學生能快速上手 Matlab 的應用，以達電腦輔助的主要目的。

基本使用語法

Matlab 中進行基本運算的方式，就是在 Matlab 命令視窗內的提示符號（">>" 或 "?"）後面輸入運算，然後按 Enter 鍵。

例 1：
`>>(7*3+5.5)/10`
`ans = 2.6500`

上例中，若不希望 Matlab 在每次運算後都顯示結果，則在運算是後面加上 ";" 即可。

`>>(7*3+5.5)/10;`
 查詢運算結果就在提示符號『>>』後面鍵入 "ans" 即可。
`ans = 2.6500`

另外，使用者也可以將運算結果儲存在自訂的變數中。

例 2：
`>>n = (7*3+5.5)/10`
`n = 2.6500`

註：MATLAB 變數字母大小寫有區別。

指令	內容	範例
help	說明	
clc	清除工作視窗螢幕	
clf	清除圖形	
clear	清除記憶的變數值	
lookfor	關鍵字搜尋	
display	顯示	display('Hello')
input	輸入	a=Input('a:')

		name=Input('姓名：','s')
load	呼叫檔案	
rand	產生 0~1 的亂數	
fix	無條件捨去	fix(4.56) ans=4
round	四捨五入	round(4.56) ans=5

如果要一次輸入的運算式在一個以上，我們可以用 "," 或 ";" 來隔開。

例 3：>>n = (7*3+5.5)/10; p = n ^ 5
p = 130.6861

若是運算式太長的話，則可以用三個句號 "..." 隔開延伸到下一行

例 4：>>(7*3+5.5)/10*...
50^5
ans = 828125000

在運算式中可以 "%" 符號加入文字，當作運算式的註解，運用在 Matlab 程式撰寫中可提高可讀性。

例 5：>>m = (7*3+5.5)/ 10 %將運算結果儲存在變數 m 中
m = 2.6500

Matlab 中有一些永久常數（Permanent Constants）供使用者直接呼叫。

例 6：>>pi
ans = 3.1416

常用到的幾個 Matlab 永久常數：

常數名稱	常數代表意義
i、j	$\sqrt{-1}$
eps	代表電腦無法分析非常小的數
inf	無限大
nan 或 NaN	無法定義的數值（Not A Number）
pi	圓周率 $\pi = 3.1415926...$

一般數學運算

我們可以透過 Matlab 中的內建函數執行一般的數學運算。

例 1：>>y=sin(50)*exp(-0.9*2^3)
y = -1.9589e-004

常用基本函數

函數：有輸入值和回傳值

常用基本數學運算函數：

函數名稱	函數功能
abs(x)	求純量或矩陣的絕對值或向量的長度 若 x 為複變數則求 x 的大小值
angle(z)	求複數 z 的相位角
sqrt(x)	求 x 的平方根
real(z)	求複數 z 的實部
imag(z)	求複數 z 的虛部
conj(z)	求複數 z 的共軛複數
round(x)	求 x 四捨五入後的最接近整數
fix(x)	捨去小數求對應於 x 的整數
floor(x)	求不大於 x 中所有數的最大整數
ceil(x)	求不小於 x 中所有數的最小整數
rat(x)	將實數 x 化為分數表示
rats(x)	將實數 x 化為多項分數展開
sign(x)	符號函數 (Signum function) $x < 0$, $\text{sign}(x) = -1$ $x = 0$, $\text{sign}(x) = 0$ $x > 0$, $\text{sign}(x) = 1$
rem(x,y)	求 x 除以 y 的餘數
gcd(x,y)	整數 x 和 y 的最大公因數
lcm(x,y)	整數 x 和 y 的最小公倍數
exp(x)	自然指數 (以 e 為底的指數)
pow2(x)	2 的指數
log(x)	自然對數 (以 e 為底的對數)
log2(x)	以 2 為底的對數

$\log_{10}(x)$	以 10 為底的對數																				
$\sin(x)$	正弦函數																				
$\cos(x)$	餘弦函數																				
$\tan(x)$	正切函數																				
$\arcsin(x)$	反正弦函數																				
$\arccos(x)$	反餘弦函數																				
$\arctan(x)$	反正切函數																				
$\sinh(x)$	雙曲正弦函數																				
$\cosh(x)$	雙曲餘弦函數																				
$\tanh(x)$	雙曲正切函數																				
$\operatorname{arsinh}(x)$	反雙曲正弦函數																				
$\operatorname{arcosh}(x)$	反雙曲餘弦函數																				
$\operatorname{artanh}(x)$	反雙曲正切函數																				
$\operatorname{mvnrnd}(\mu, \sigma, nobs)$	<p>產生多變量常態分配隨機樣本 例:</p> <pre>?mvnrnd([1;2],[1 .5; .5 1],10)</pre> <p>ans =</p> <table> <tbody> <tr><td>0.5674</td><td>1.6220</td></tr> <tr><td>-0.6656</td><td>1.7958</td></tr> <tr><td>1.1253</td><td>1.5532</td></tr> <tr><td>1.2877</td><td>4.0345</td></tr> <tr><td>-0.1465</td><td>1.3086</td></tr> <tr><td>2.1909</td><td>2.6941</td></tr> <tr><td>2.1892</td><td>3.5184</td></tr> <tr><td>0.9624</td><td>2.0325</td></tr> <tr><td>1.3273</td><td>2.0808</td></tr> <tr><td>1.1746</td><td>1.3665</td></tr> </tbody> </table>	0.5674	1.6220	-0.6656	1.7958	1.1253	1.5532	1.2877	4.0345	-0.1465	1.3086	2.1909	2.6941	2.1892	3.5184	0.9624	2.0325	1.3273	2.0808	1.1746	1.3665
0.5674	1.6220																				
-0.6656	1.7958																				
1.1253	1.5532																				
1.2877	4.0345																				
-0.1465	1.3086																				
2.1909	2.6941																				
2.1892	3.5184																				
0.9624	2.0325																				
1.3273	2.0808																				
1.1746	1.3665																				
factor	<p>因式分解 例: $\operatorname{factor}(x^2-1)$ $\text{ans} = (x-1)*(x+1)$</p>																				

繪圖

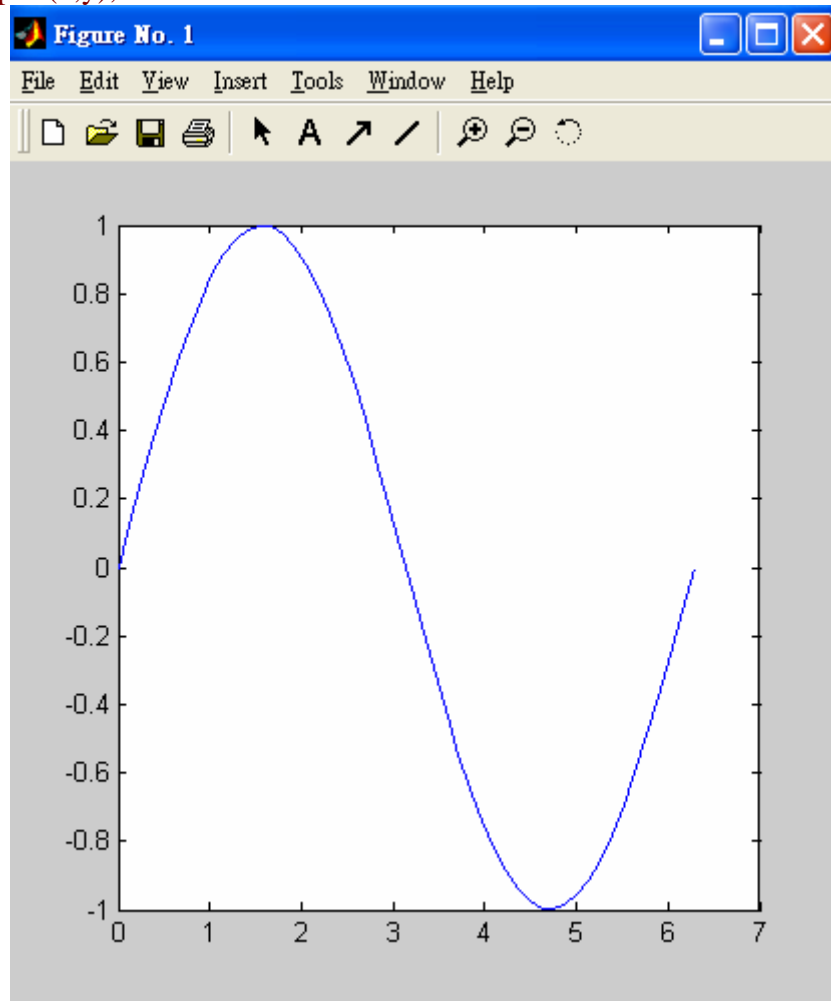
這裡介紹 Matlab 繪製曲線、二維曲面的方法以及儲存、列印的用法。

基本繪圖

`plot` 是 Matlab 繪製圖形的基本函數，透過各座標點的定義，再呼叫使用 `plot` 就可以完成簡易的一維圖形。

例：

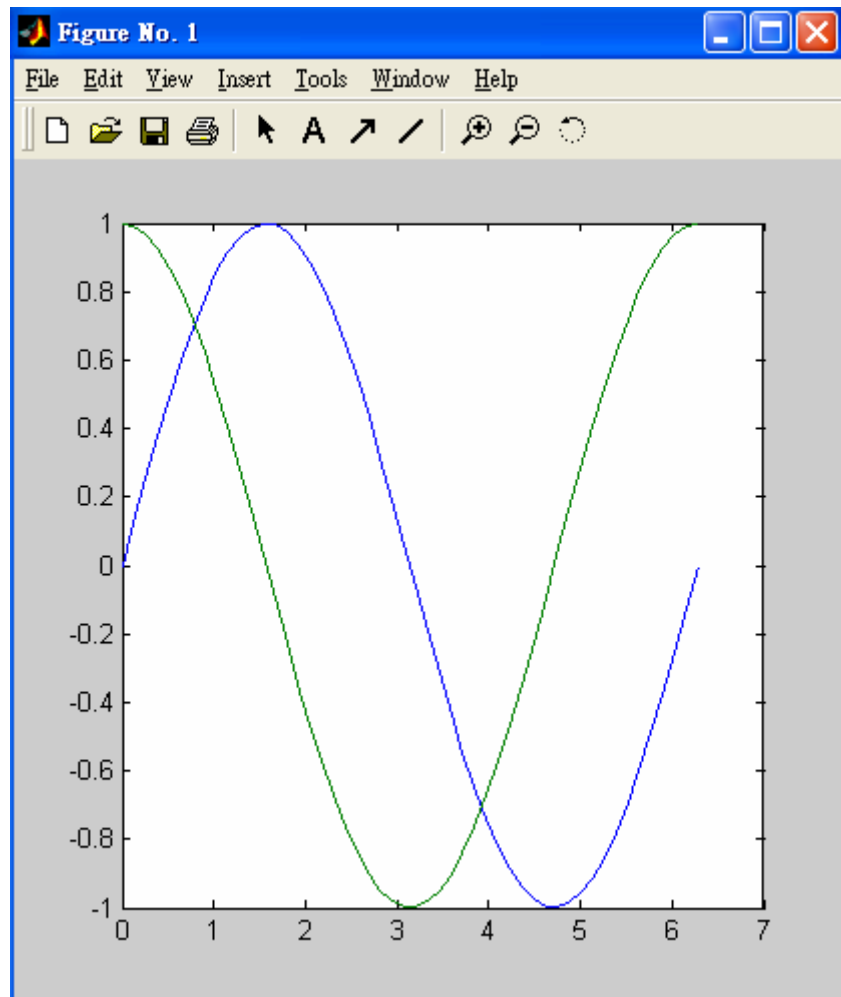
```
>> x=linspace(0, 2*pi, 50); %linspace(0, 2*pi, 50)表示 0 到 2pi 分成 50 等分  
>> y=sin(x);  
>> plot(x,y);
```



使用 `plot` 除了如上述繪製方式外，我們可以用 `plot` 來繪製多個圖形。

例：同時繪製 $y=\sin(x)$ 和 $y=\cos(x)$ 之圖形

```
>> x=linspace(0, 2*pi, 50);  
>> plot(x, sin(x), x, cos(x));
```

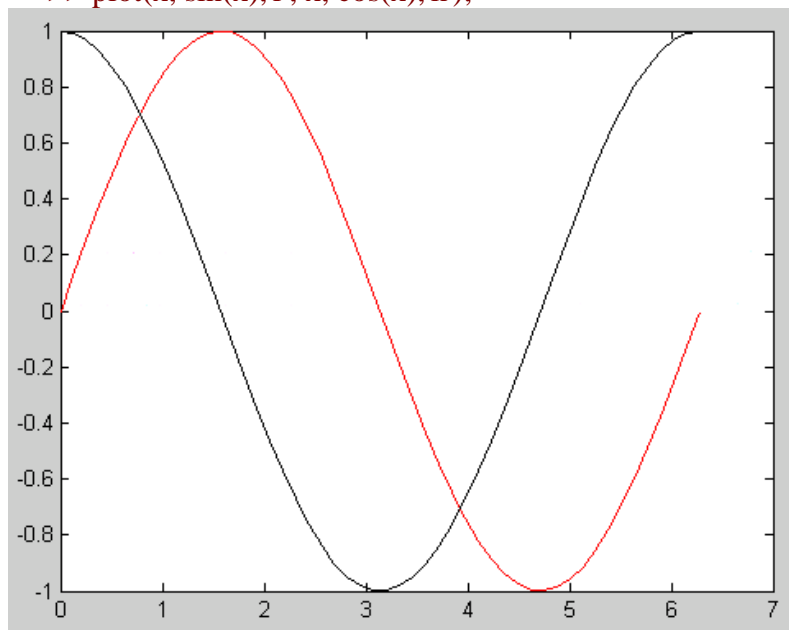


plot 繪製圖形功能中容許使用者改變顏色以及圖形樣式，可以選擇的顏色以及圖形樣式如下表：

字元	代表顏色	圖形樣式符號	代表圖形樣式
y	黃色	.	點
m	洋紅色	O	圓圈
c	青色	x	『X』符號
r	紅色	+	『+』符號
g	綠色	*	『*』符號
b	藍色	-	實線
w	白色	:	點線
k	黑色	--	虛線

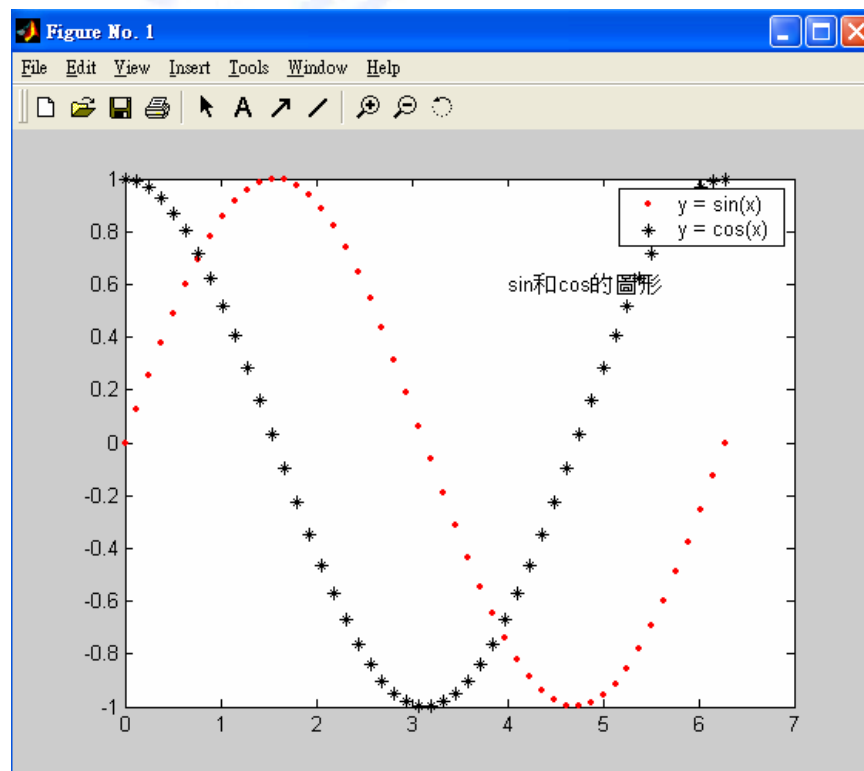
例：plot 改變顏色

```
>> x=linspace(0, 2*pi, 50);
>> plot(x, sin(x),'r', x, cos(x),'k');
```



例：改變 plot 圖形樣式

```
>> x=linspace(0, 2*pi, 50);
>> plot(x, sin(x),'r.', x, cos(x),'k*');
>> legend('y = sin(x)', 'y = cos(x)'); % 圖形註解
>> text(4,0.6,'sin 和 cos 的圖形') %text：圖中文字說明
```

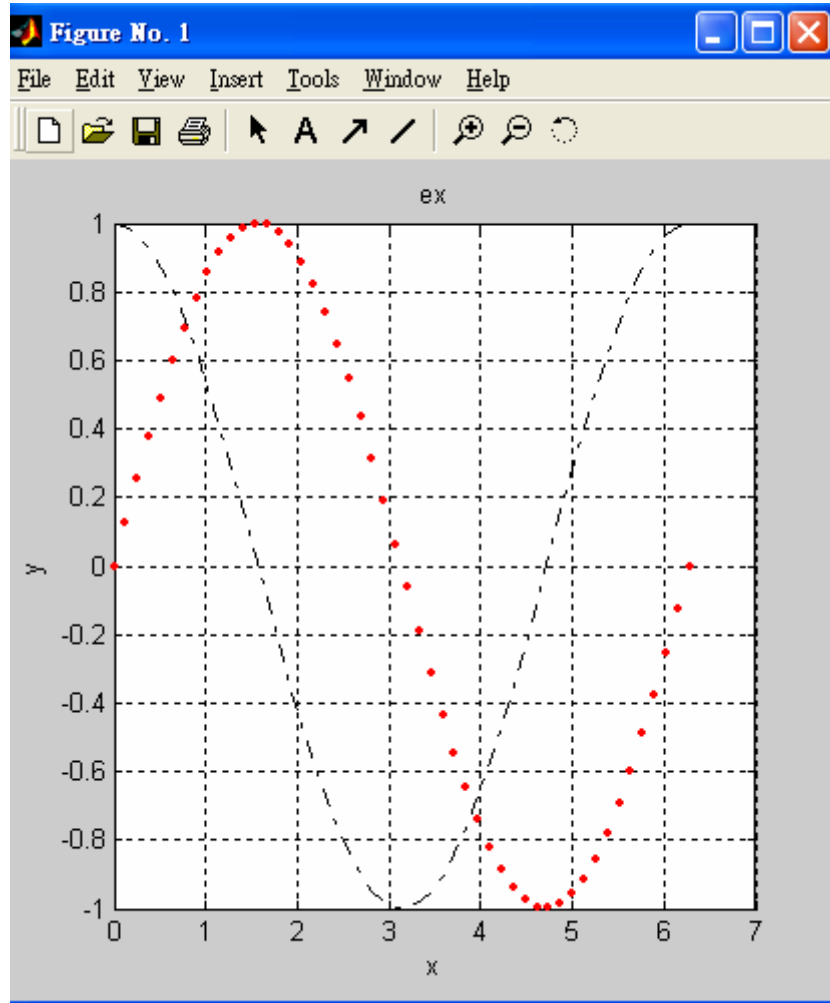


圖形繪製中為了增加圖形的可讀性，通常會在圖形中加上圖形註解。

若要在圖形中加入圖形註解，我們可以使用『xlabel』、『ylabel』、『title』、『legend』、『grid on』、『text』

例：

```
>>x=linspace(0, 2*pi, 50);
>>plot(x, sin(x),'r.',x, cos(x),'k-.');
>>xlabel('x');           % x 軸註解
>>ylabel('y');           % y 軸註解
>>title('ex');           % 圖形標題
>>grid on;               % 顯示格線
```

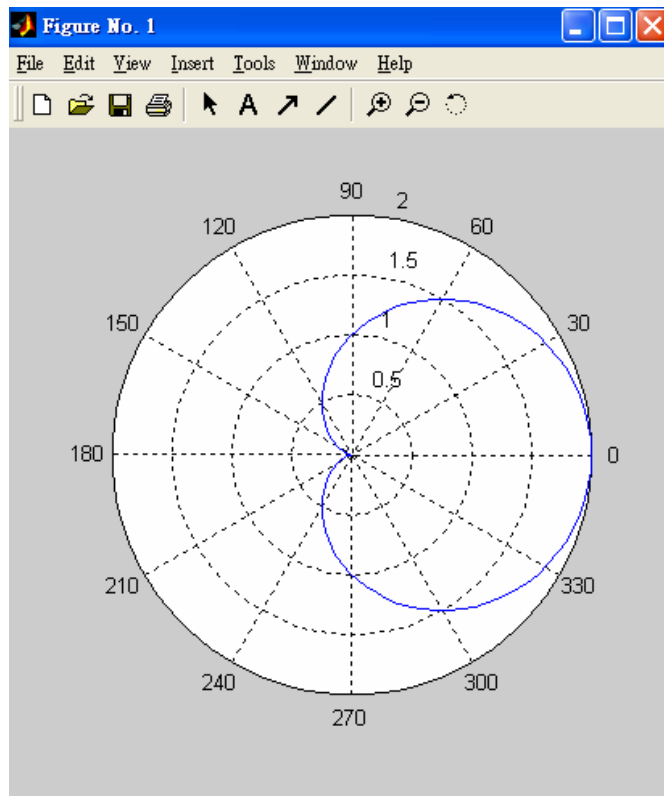


二維圖形

Matlab 有提供極座標圖形的函數如下。

例：polar () 極座標圖

```
>> theta=linspace(0, 2*pi);
>> r=1+cos(theta);
>> polar(theta, r);
```

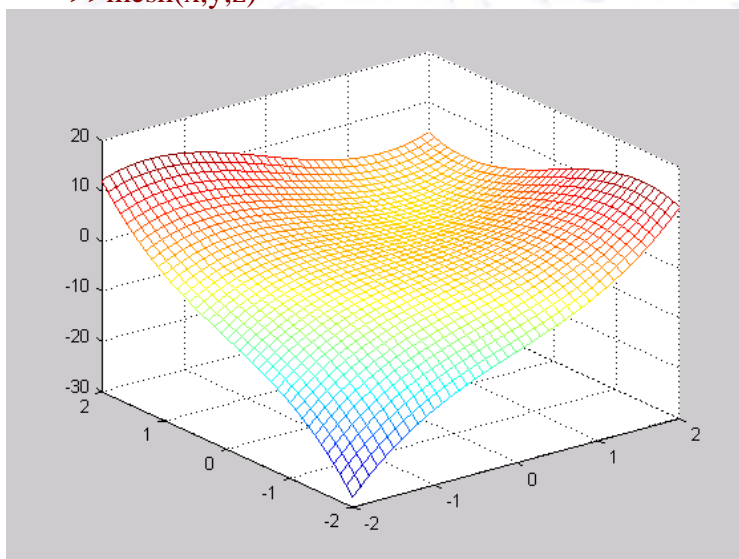


三維圖形

Matlab 中的三維空間繪圖基本函數為『mesh』、『plot3』，其中 mesh 可以繪製出立體網狀圖，而 plot3 可以繪製出立體曲線圖，所繪製出的圖形會以不同的顏色來代表不同高度。

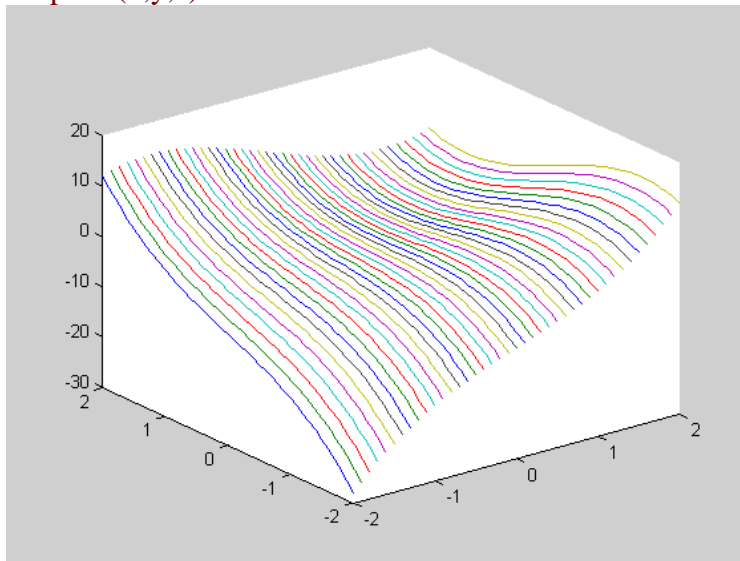
例十一：立體網狀圖 $z = x^3 + y^3 - 3xy$

```
>> [x,y]=meshgrid(-2:0.1:2);
>> z=x.^3+y.^3-3*x.*y;
>> mesh(x,y,z)
```



例十二： $z = x^3 + y^3 - 3xy$

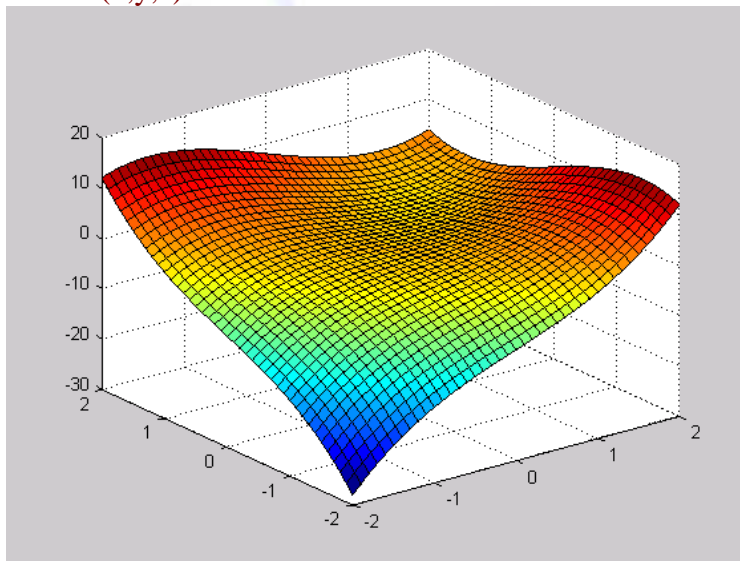
```
>> [x,y]=meshgrid(-2:0.1:2);
>> z=x.^3+y.^3-3*x.*y;
>> plot3(x,y,z)
```



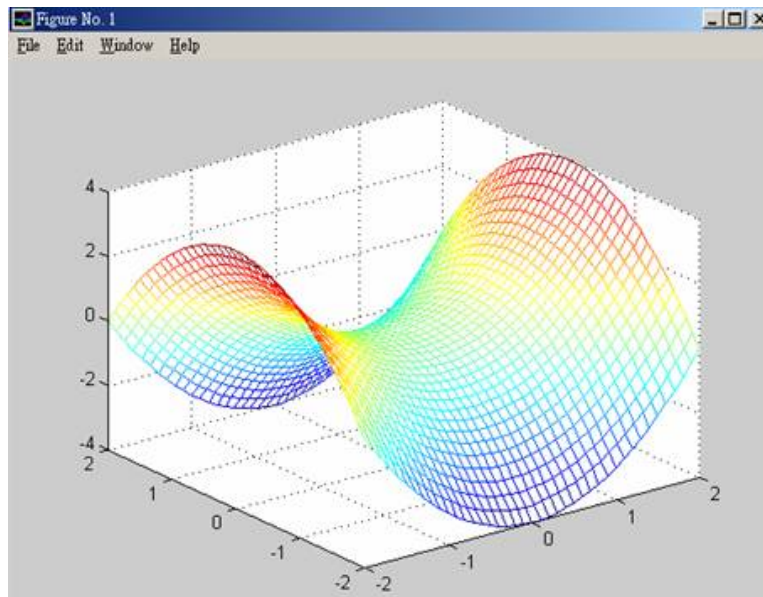
另外一個 surf 用法類似 mesh，介紹如下：

例十三：立體曲面圖 $z = x^3 + y^3 - 3xy$

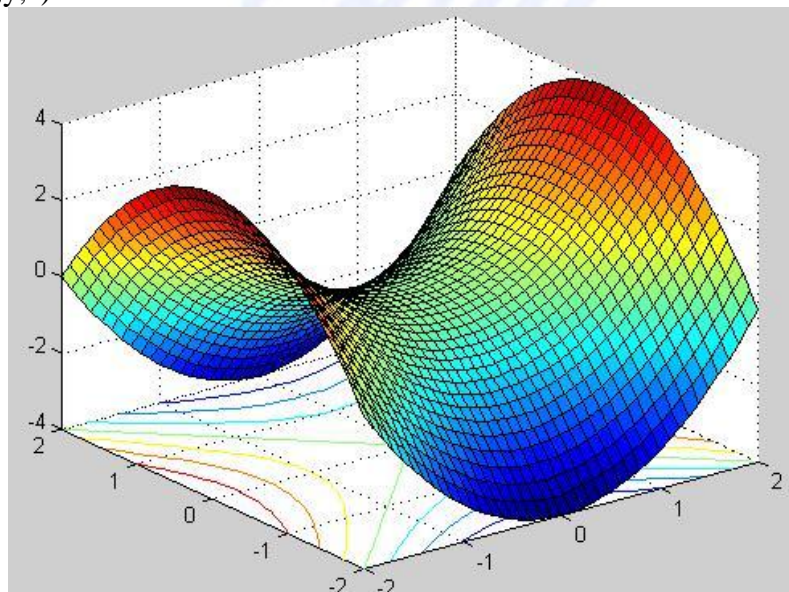
```
>> [x,y]=meshgrid(-2:0.1:2);
>> z=x.^3+y.^3-3*x.*y;
>> surf(x,y,z)
```



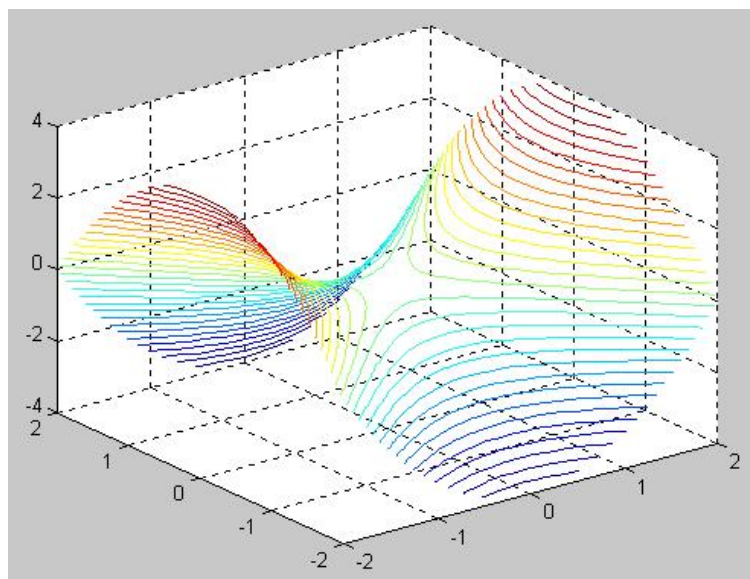
```
[x,y]=meshgrid(-2:0.1:2);
z=x.^2-y.^2;
mesh(x,y,z)
```



%meshc(同時畫出網狀圖與等高線) & surfc(可同時畫出曲面圖與等高線)
 [x,y]=meshgrid(-2:0.1:2);
 z=x.^2-y.^2;
 surfc(x,y,z)



%contour(二維等高線),contour3(三維等高線)
 [x,y]=meshgrid(-2:0.1:2);
 z=x.^2-y.^2;
 contour3(x, y, z, 30);



矩陣基本運算

矩陣基本運算

一個 3 x 3 矩陣

```
1 2 3
4 5 6
7 8 9
```

直接在 Matlab 命令視窗內的提示符號『>>』後面輸入，各項元素間至少以一個空格來分隔，列與列間則以";"分隔。而指令行最後加上個分號可避免顯示出許多不需要的東西。

例 6：>>A=[1 2 3;4 5 6;7 8 9];

若想知道矩陣A之內容時，則直接提示符號『>>』後面輸入『A』即可，螢幕之顯示結果如下：

```
A=
1 2 3
4 5 6
7 8 9
```

另一種輸入可以不加分號，但是強迫分行，亦可達到相同的效果：

例 7：

```
>> B=[1 3
5 7]
```

螢幕之顯示結果如下：

```
B=
1 3
5 7
```


特殊矩陣

ones(3,4)：元素全部是 1 的 3×4 階矩陣

eye(3)：3 階單位矩陣

A(2,3)：第 2 列第 3 行的元素

A(2,:)：第 2 列的所有元素

A(:,3)：第 3 行的所有元素

Matlab 中矩陣的基本運算符號有以下這些

+	：加法運算	-	：減法運算
*	：乘法運算	/	：右除法運算
\	：左除法運算	^	：次方運算
'	：轉置矩陣		

例 8：（基本運算）

轉置矩陣：

```
>>B'
```

```
ans =
```

```
1 5
```

```
3 7
```

乘法運算：

```
>>B*B
```

```
ans =
```

```
16 24
```

```
40 64
```

次方運算：

```
>>B^3
```

```
ans =
```

```
856576 1353216
```

```
2255360 3563008
```

矩陣運算中所要注意的是『矩陣的左、右除法運算結果不相等』。

例 9：

```
>>C=[7 9;11 13];
```

```
>>B/C
```

```
ans =
```

```
2.5000 -1.5000
```

```
1.5000 -0.5000
```

```
>>B\C
```

```
ans =
```

```
-2 -3
```

```
3 4
```

Matlab 中也可以進行複變數的宣告與運算，以 "i" 或 "j" 還代表虛數，如 $y=2+3i$ 。

Matlab 中有幾個函數可以產生矩陣，例如：rand。

例 10：rand(n) 與 rand(m,n) 將個別產生 $n \times n$ 與 $m \times n$ 亂數矩陣，元素分佈於 0 與 1 之間。

```
>>rand(3)
ans =
0.9501 0.4860 0.4565
0.2311 0.8913 0.0185
0.6068 0.7621 0.8214
```

A=[1 3;2 4]

```
A = 1 3
     2 4
```

• A'

```
ans = 1 2
      3 4
```

• det(A)

```
ans = -2
```

• inv(A)

```
ans = -2.0000 1.5000
      1.0000 -0.5000
```

• A^2-2*A+eye(2)

```
ans = 6 9
      6 15
```

• eig(A)

```
ans = -0.3723
      5.3723
```

• [P,D]=eig(A)

```
P = -0.9094 -0.5658
      0.4160 -0.8246
D = -0.3723 0
      0 5.3723
```

常用矩陣運算

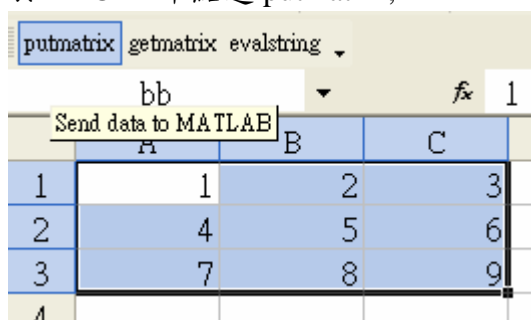
名稱	功能
A.^2	每個元素平方
A^2	矩陣平方運算

inv(A)	求矩陣 A 的逆矩陣 (inverse)
det(A)	求矩陣 A 的行列式值
log(A)	每個元素取 log
eig(A)	求矩陣 A 的特徵值與特徵向量
sum(A)	行加總
sum(sum(A))	全部元素加總
std(A)	計算行的標準差

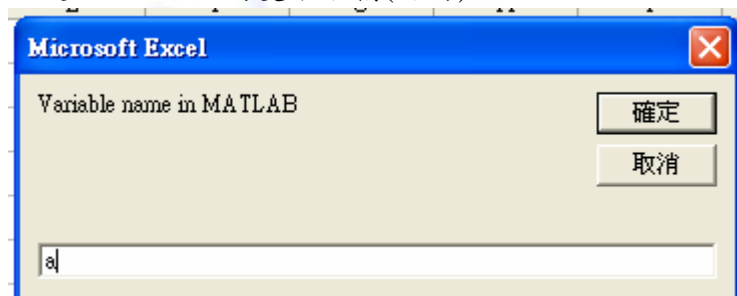
和 excel 互傳資料

點選 MATLAB toolbox 之 exlink

於 EXCEL 中點選 putmatrix,



給定 MATLAB 的變數名稱(自訂)



於 MATLAB 中呼叫此變數即可得

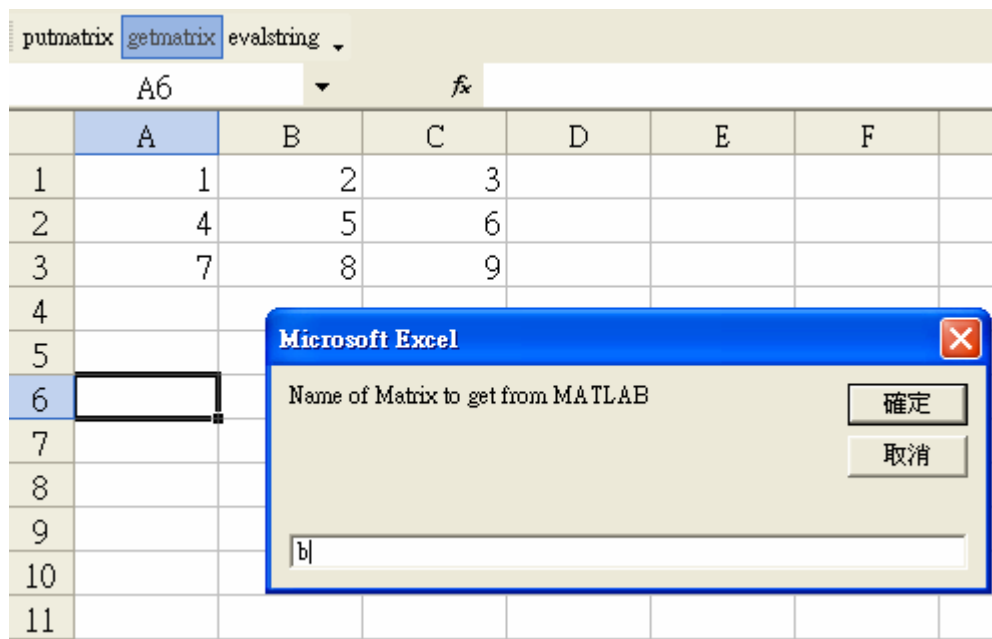
```
?a
a =
     1     2     3
     4     5     6
     7     8     9
```

於 MATLAB 中執行 b=a+1(即 b 矩陣之每個元素加 1)

```
?b=a+1
```

```
b =
     4     5     6
     7     8     9
    10    11    12
```

於 EXCEL 中點選 getmatrix,



可得

	A	B	C
1	1	2	3
2	4	5	6
3	7	8	9
4			
5			
6	4	5	6
7	7	8	9
8	10	11	12
9			

微積分應用

Matlab 在微積分可求極限、導函數及積分等運算。

例一：求極限

syms x y a t h; %syms 定義符號變數

- `limit(sin(x)/x)`
`ans = 1`
- `limit((x-2)/(x^2-4),2)`
`ans = 1/4`
- `limit((1+2*t/x)^(3*x),x,inf)`
`ans = exp(6*t)`

- `limit(1/x,x,0,'right')`
`ans =inf`
- `limit(1/x,x,0,'left')`
`ans =-inf`
- `limit((sin(x+h)-sin(x))/h,h,0)`
`ans =cos(x)`
- `v = [(1 + a/x)^x, exp(-x)];`
`limit(v,x,inf,'left')`
`ans =[exp(a), 0]`

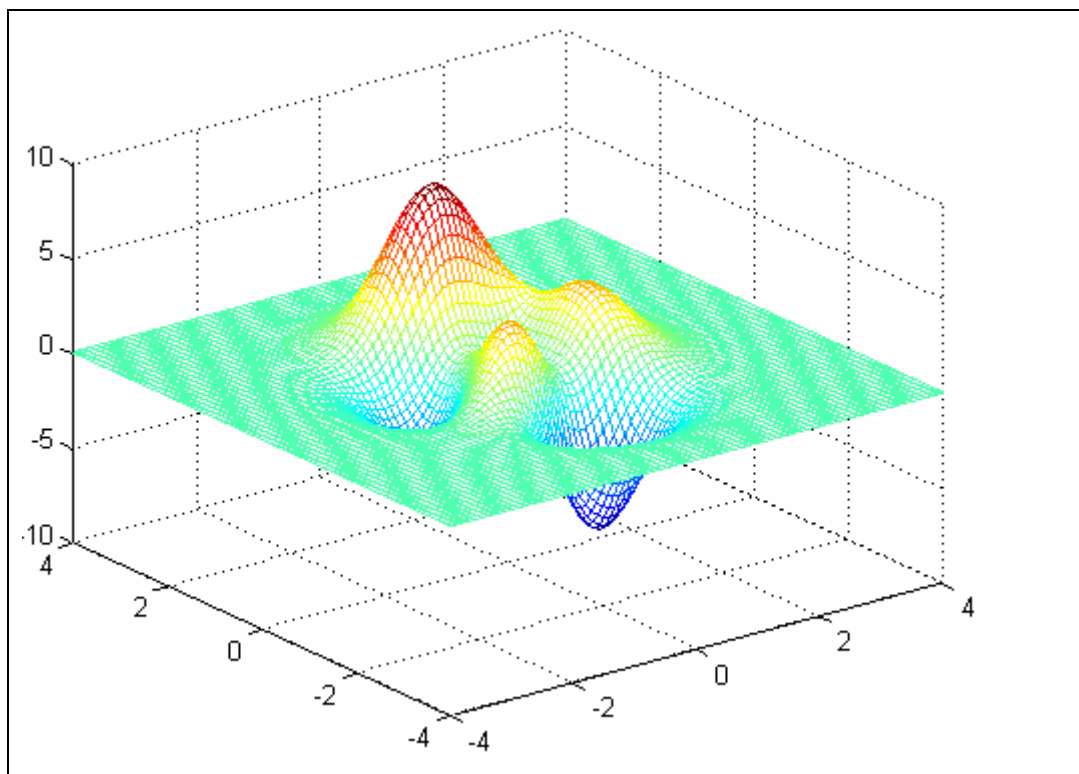
例二：求導函數

syms x y; %syms 定義符號變數

- % 求 $y=3x^4+2x^3-x+7$ 之導函數
`diff('3*x^4+2*x^3-x+7')`
`ans=12*x^3+6*x^2-1`
- % 求 $y=(x^3+x^2+2)(2x^3-x^2+3)$ 之導函數
`diff('(x^3+x^2+2)*(2*x^3-x^2+3)')`
`ans=(3*x^2+2*x)*(2*x^3-x^2+3)+(x^3+x^2+2)*(6*x^2-2*x)`
- % 求 $y=x/x^{1/2}$ 之導函數
`diff('x/sqrt(x)')`
`ans=1/2/x^(1/2)`
- % 求 $y=\sec^4(x)$ 之導函數
`diff('sec(x)^4')`
`ans=4*sec(x)^4*tan(x)`
- % 求 $z=3*(1-x)^2*\exp(-(x^2)-(y+1)^2)-10*(x/5-x^3-y^5)*\exp(-x^2-y^2)-1/3*\exp(-(x+1)^2-y^2)$ 之一階和二階偏導函數

圖形

```
>> [x,y]=meshgrid(-4:0.1:4);
>> 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);
>> mesh(x,y,z)
```



```
>> syms x y
>> 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);
>> diff(z,x) (求  $\frac{\partial z}{\partial x}$ )
ans = -6*(1-x)*exp(-x^2-(y+1)^2)-6*(1-x)^2*x*exp(-x^2-(y+1)^2)
    -(2-30*x^2)*exp(-x^2-y^2)+2*(2*x-10*x^3-10*y^5)*x*exp(-x^2-y^2)
    -1/3*(-2*x-2)*exp(-(x+1)^2-y^2)
>> diff(z,y) (求  $\frac{\partial z}{\partial y}$ )
ans = 3*(1-x)^2*(-2*y-2)*exp(-x^2-(y+1)^2)+50*y^4*exp(-x^2-y^2)
    +2*(2*x-10*x^3-10*y^5)*y*exp(-x^2-y^2)+2/3*y*exp(-(x+1)^2-y^2)
>> diff(diff(z,x),x) (求  $\frac{\partial^2 z}{\partial x^2}$ )
ans = 6*exp(-x^2-(y+1)^2)+24*(1-x)*x*exp(-x^2-(y+1)^2)
    -6*(1-x)^2*exp(-x^2-(y+1)^2)+12*(1-x)^2*x^2*exp(-x^2-(y+1)^2)
    +60*x*exp(-x^2-y^2)+4*(2-30*x^2)*x*exp(-x^2-y^2)+2*(2*x-10*x^3
    -10*y^5)*exp(-x^2-y^2)-4*(2*x-10*x^3-10*y^5)*x^2*exp(-x^2-y^2)
    +2/3*exp(-(x+1)^2-y^2)-1/3*(-2*x-2)^2*exp(-(x+1)^2-y^2)
>> diff(diff(z,x),y) (求  $\frac{\partial^2 z}{\partial x \partial y}$ )
ans = -6*(1-x)*(-2*y-2)*exp(-x^2-(y+1)^2)-6*(1-x)^2*x*(-2*y-2)
    *exp(-x^2-(y+1)^2)+2*(2-30*x^2)*y*exp(-x^2-y^2)-100*y^4*x
    *exp(-x^2-y^2)-4*(2*x-10*x^3-10*y^5)*x*y*exp(-x^2-y^2)
    +2/3*(-2*x-2)*y*exp(-(x+1)^2-y^2)
```

```
>> diff(diff(z,y),x) (求  $\frac{\partial^2 z}{\partial y \partial x}$ )
ans = -6*(1-x)*(-2*y-2)*exp(-x^2-(y+1)^2)-6*(1-x)^2*x*(-2*y-2)
      *exp(-x^2-(y+1)^2)+2*(2-30*x^2)*y*exp(-x^2-y^2)-100*y^4*x
      *exp(-x^2-y^2)-4*(2*x-10*x^3-10*y^5)*x*y*exp(-x^2-y^2)
      +2/3*(-2*x-2)*y*exp(-(x+1)^2-y^2)
>> diff(diff(z,y),y) (求  $\frac{\partial^2 z}{\partial y^2}$ )
ans = -6*(1-x)^2*exp(-x^2-(y+1)^2)
      +3*(1-x)^2*(-2*y-2)^2*exp(-x^2-(y+1)^2)
      +200*y^3*exp(-x^2-y^2)-200*y^5*exp(-x^2-y^2)
      +2*(2*x-10*x^3-10*y^5)*exp(-x^2-y^2)
      -4*(2*x-10*x^3-10*y^5)*y^2*exp(-x^2-y^2)
      +2/3*exp(-(x+1)^2-y^2)-4/3*y^2*exp(-(x+1)^2-y^2)
```

Matlab 中以 int 指令作積分運算

例三：積分運算（定積分）

```
syms x y; %syms 定義符號變數
int('x^2+1','x',0,1)
ans = 4/3
```

```
int('1/sqrt(4-x^2)','x',0,1)
ans = 1/12*4^(1/2)*pi
```

例四：積分運算（不定積分）

```
syms x y; %syms 定義符號變數
int(x)
ans = 1/2*x^2
```

```
int(sin(x))
ans = -cos(x)
```

例五：泰勒級數

```
syms x y; %syms 定義符號變數
taylor(1/(1-x))
ans = 1+x+x^2+x^3+x^4+x^5
```

```
taylor(exp(x))
ans = 1+x+1/2*x^2+1/6*x^3+1/24*x^4+1/120*x^5
```

```
taylor(sin(x))
ans = x-1/6*x^3+1/120*x^5
```

```
taylor(cos(x))
ans = 1-1/2*x^2+1/24*x^4
```

```
taylor(cos(x),10)
```

```
ans= 1-1/2*x^2+1/24*x^4-1/720*x^6+1/40320*x^8
```

```
taylor(log(1+x))
```

```
ans= x-1/2*x^2+1/3*x^3-1/4*x^4+1/5*x^5
```

```
taylor(exp(-x))
```

```
ans=1-x+1/2*x^2-1/6*x^3+1/24*x^4-1/120*x^5
```

```
taylor(log(x),6,1)
```

```
ans=x-1-1/2*(x-1)^2+1/3*(x-1)^3-1/4*(x-1)^4+1/5*(x-1)^5
```

```
taylor(sin(x),pi/2,6)
```

```
ans=1-1/2*(x-1/2*pi)^2+1/24*(x-1/2*pi)^4
```

程式撰寫

Matlab 對於高階應用需求提供程式撰寫的功能，可用來處理大量資料的計算或系統模擬。這樣的方式很類似過去作業系統 DOS 中撰寫的批次檔一般，把許多一連串指令或函數集合起來運作。

通常 MATLAB 程式碼撰寫的檔案以『m』為副檔名。這些副檔名為『m』的檔案一般又分為兩種類型，其中一種為巨集式的 MATLAB 命令集；另一種則是構成函式(function)。使用者可以使用 MATLAB 內建的『M 檔案編輯器』(M-File Editor)

或其他編輯器來撰寫編輯 M 檔案。

檔案命名：

1. 不要用 MATLAB 已有之函數名稱
2. 不要用特殊符號
3. 文字在前，數字在後

底下為一個 M 檔案內容範例：（今檔名為 sample1.m）

```
%程式撰寫繪圖(碎形)
```

```
clear all
```

```
hold
```

```
a=[0 0 0.5 0 0.27 0 -0.139 0.263 0.57 0.246 0.224 -0.036 0.17 -0.215 0.408 0.222  
0.176 0.0893;0.781 0.034 0.1075 -0.032 0.739 0.27];
```

```
v=0;
```

```
w=0;
```

```
for l=1:10000
```

```
  r=rand;
```

```
  if r<0.1
```

```
    k=1;
```

```
  elseif r<0.2
```

```
    k=2;
```

```
  elseif r<0.3
```

```
    k=3;
```

```
  else
```

```
    k=4;
```

```
  end
```



```

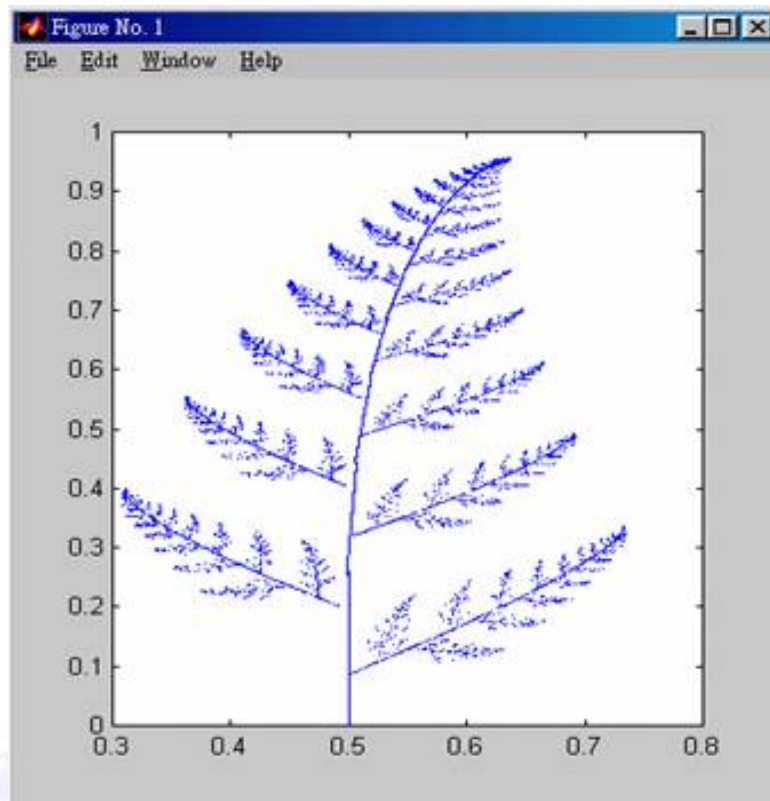
z=v*a(k,1)+w*a(k,2)+a(k,3);
u=v*a(k,4)+w*a(k,5)+a(k,6);
plot(z,u)
v=z;
w=u;
end

```

在 MATLAB 中我們要執行這個 M 檔案中的程式碼，可以在 MATLAB 的指令視窗中輸入

```
>>sample1
```

則會出現程式執行結果



像 sample1.m 這類的 M 檔案所產生的效果，等同於在 MATLAB 指令列中一行一行輸入 sample1.m 裡每一列指令所產生的結果。這類型的 M 檔案即為巨集式 M 檔案。

```

%程式撰寫繪圖(碎形) sample2.m
clear all
hold
a=[.5 0 0 .5 0;.5 0 .5 0 .5 0;.5 0 0 0 .5 .5];
v=0;
w=0;
for l=1:10000
r=rand;
if r<0.3
k=1;
elseif r<0.6
k=2;
else

```

```

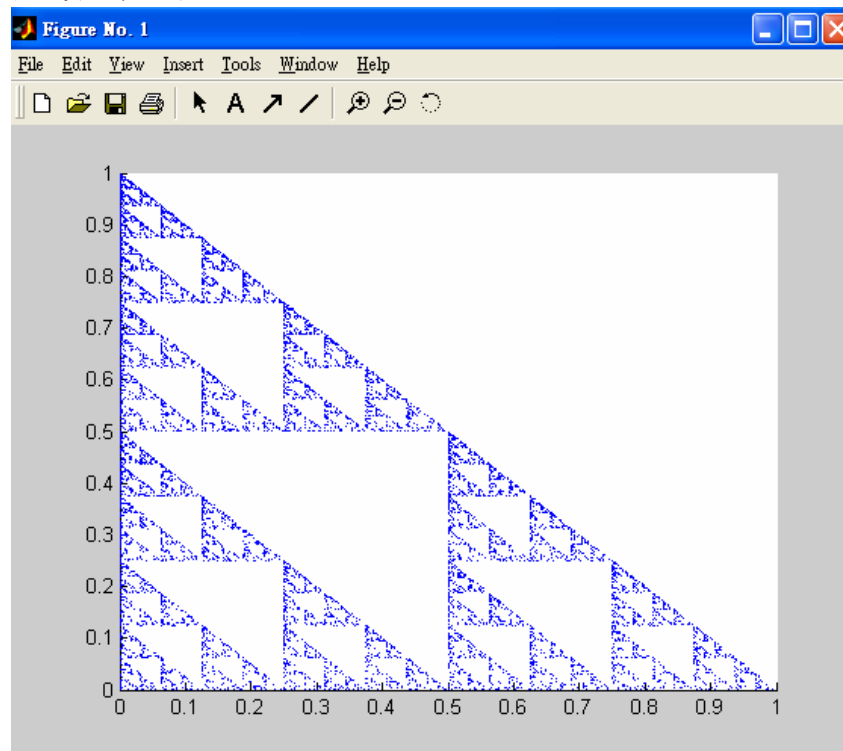
k=3;
end
z=v*a(k,1)+w*a(k,2)+a(k,3);
u=v*a(k,4)+w*a(k,5)+a(k,6);
plot(z,u)
v=z;
w=u;
end

```

在 MATLAB 中我們要執行這個 M 檔案中的程式碼，可以在 MATLAB 的指令視窗中輸入

```
>>sample2
```

則會出現程式執行結果



% 程式撰寫繪圖(碎形) sample3.m

```

clear all
hold
a=[1/3 0 0 0 1/3 0;1/3 0 2/3 0 1/3 0;1/3 0 1/3 0 1/3 1/3;1/3 0 0 0 1/3 2/3;1/3 0 2/3 0
1/3 2/3];
v=0;
w=0;
for l=1:10000
r=rand;
if r<0.2
k=1;
elseif r<0.4
k=2;
elseif r<0.6
k=3;
elseif r<0.8

```

```

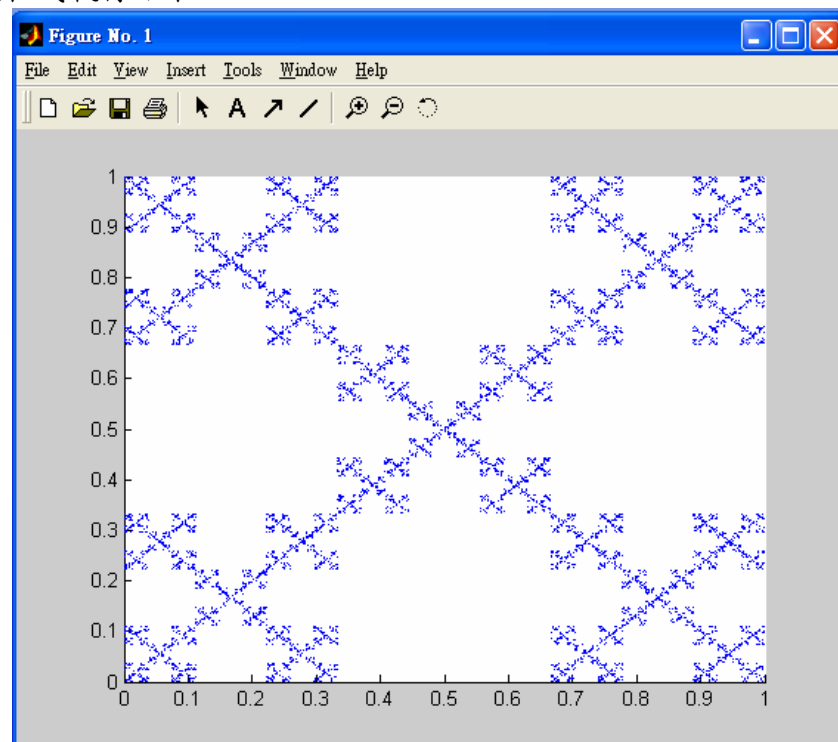
k=4;
else
k=5;
end
z=v*a(k,1)+w*a(k,2)+a(k,3);
u=v*a(k,4)+w*a(k,5)+a(k,6);
plot(z,u)
v=z;
w=u;
end

```

在 MATLAB 中我們要執行這個 M 檔案中的程式碼，可以在 MATLAB 的指令視窗中輸入

>>sample3

則會出現程式執行結果



程式流程控制

MATLAB 在程式撰寫應用中有一些指令會影響程式執行的流程，可以歸納為兩類，

一種為迴圈指令，另一種則是條件指令。

MATLAB 迴圈類指令有 for 迴圈 以及 while 迴圈。

for 迴圈

for 迴圈一般用法

for varname = x:y:z

例：

```

for i=1:1:10    % for i = 起始數值:遞增數值:終止數值
    for j=1:1:10

```

```

    a(i,j)=i+j;
end
end

```

上面 sample1.m 的範例中也有 for 迴圈

for 迴圈一般是用來處理已知執行次數的固定重複運算或數值設定工作。

在 for 迴圈的運用中特別注意的是要在景 r 迴圈結尾加上『end』，否則 for 將不會終止。

例：利用矩形中點法、梯形法、Simpson 法 求 $f(x)=\sqrt{x}\sin(x)$ 在區間 $[0, 4]$ 分割 $n=8$ 等分時的數值積分。

自訂函數

```
function [v,w]=f(x,y)
```

```
v=x^2+y^2;
```

```
w=x+y;
```

```
[v,w]=f(4,-1)
```

```
v =17
```

```
w =3
```

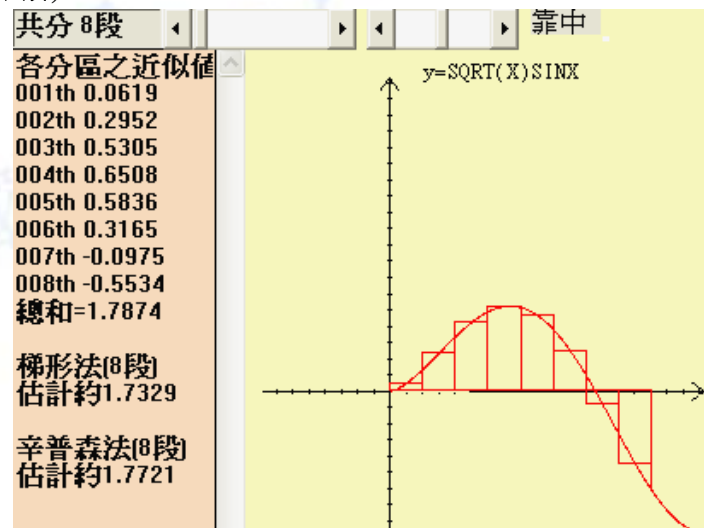
f.m

(自訂函數)

```
function [value]=f(x)
```

```
value=sqrt(x)*sin(x);
```

m.m (矩形中點法)



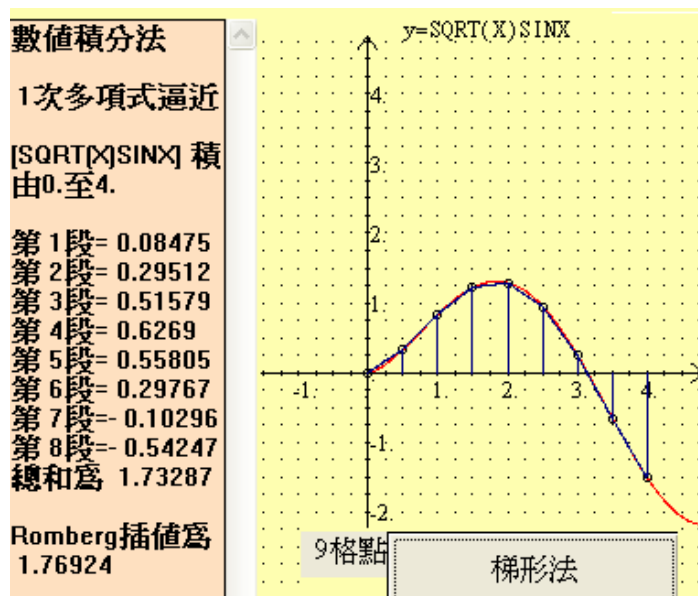
```

a=0;
b=4;
n=8;
h=(b-a)/n;
s1=0;
sum=0;
for i=0 :n-1;
s1=f(a+(i+1/2)*h);
sum=sum+s1;
end

```

sum*h
ans =1.7874

t.m (梯形法)

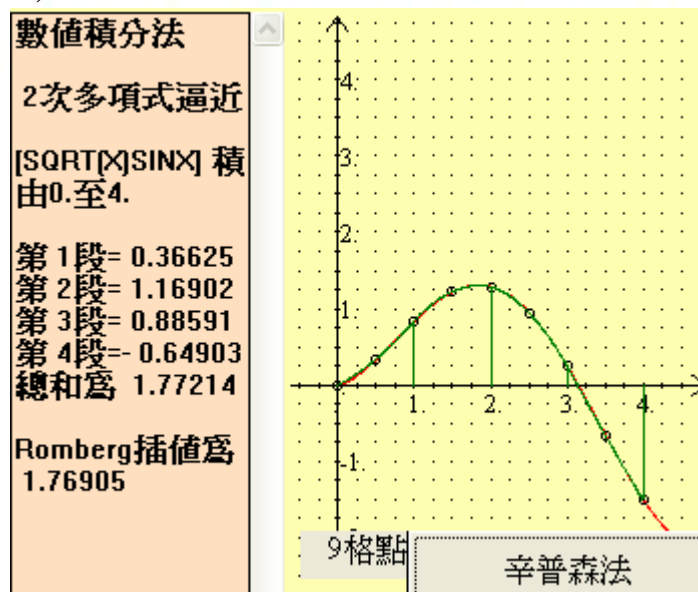


```

a=0;
b=4;
n=8;
h=(b-a)/n;
s1=0;
sum=0;
for i=0 :n-1;
s1=f(a+i*h)+f(a+(i+1)*h);
sum=sum+s1;
end
sum*h/2
ans =1.7329

```

s.m (Simpson 法)



```

a=0;
b=4;
n=8;
h=(b-a)/n;
s1=0;
sum=0;
for i=0 : 2:n-2;
s1=f(a+i*h)+4*f(a+(i+1)*h)+f(a+(i+2)*h);
sum=sum+s1;
end
sum*h/3
ans = 1.7721

```

while 迴圈

while 迴圈一般用法

```

while expression
    statements
end

```

例：

```

n=1;
while n < 98765
    n=n+1;
end

```

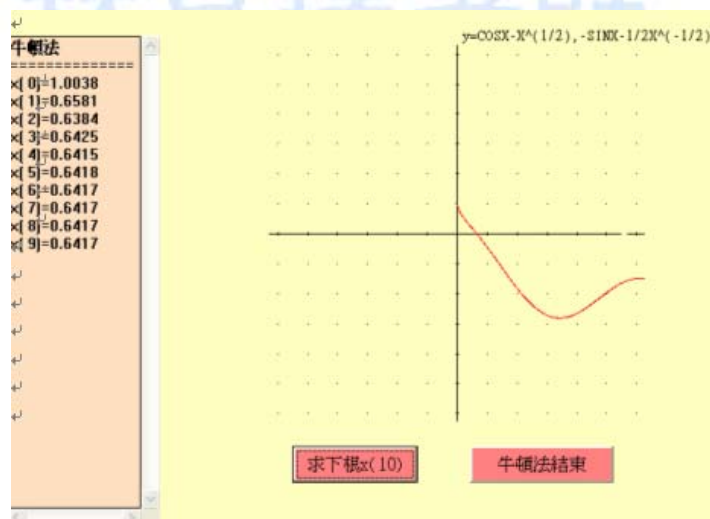
while 迴圈適用於執行次數未知的運算。

例：利用牛頓法解方程式 $\cos(x)-x^{1/2}=0$ 可用下列之語法

```

a=1
while abs(cos(a)-a^(1/2))>0.0001
b=a-(cos(a)-a^(1/2))/(-sin(a)-1/2*a^(-1/2))
a=b
end
a
a =
    0.6417

```



```

例： 利用二分法解方程式  $e^x - 3x^2 = 0$ 
y=exp(x)-3*x.^2
xl=3; xr=4 %由圖形觀測而得之初估值
format long
while abs(xr-xl)>10^(-6)
    yold=exp(xl)-3*xl.^2
    xtmp=(xl+xr)/2; ynew=exp(xtmp)-3*xtmp.^2
    if yold*ynew<0
        xr=xtmp
    else
        xl=xtmp
    end
end
end
(xl+xr)/2

ans =
    3.73307943344116

```

MATLAB 條件類指令可分為 if-else 和 switch case

```

if
if 條件指令的一般用法
if 邏輯條件式
    運算式
end

```

if 條件指令後面的邏輯條件式裡頭的邏輯條件成立（True），則會執行 if 條件指令裡頭的運算式。

if 條件指令可以結合 else 以及 elseif 作更進一步的程式流程控制。

```

if 邏輯條件式
    運算式 A;
else
    運算式 B;
end

```

以及

```

if 邏輯條件式 A
    運算式 A;
elseif 邏輯條件式 B
    運算式 B;
elseif 邏輯條件式 C
    運算式 C;
else
    運算式 D;
end

```

例：

```
a=input('score')
if a>=80
    disp('甲等');
else if a>=70
    disp('乙等');
    else if a>=60
        disp('丙等');
else
    disp('丁等');
end
end
end
```

前面我們提過的範例 sample1.m 裡頭也有 if 條件指令。

```
switch case
switch case 的一般用法
switch 條件式
    case 值 A
        運算式 A;
    case 值 B
        運算式 B;
    .
    .
    .
end
```

例：

```
a=input('請輸入滿意程度1-5')
switch a
    case 1
        disp('非常滿意')
    case 2
        disp('滿意')
    case 3
        disp('無意見')
    case 4
        disp('不滿意')
    case 5
        disp('非常不滿意')
    otherwise
        disp('輸入錯誤')
end
```


應用範例

Laplace

- syms t
laplace(t)
ans = 1/s^2
- syms s
ilaplace(1/s^2)
ans = t

Fourier

- syms x
fourier(exp(-x^2))
ans = pi^(1/2)*exp(-1/4*w^2)
- syms w
ifourier(pi^(1/2)*exp(-1/4*w^2))
ans = 3991211251234741/4503599627370496*4^(1/2)/pi^(1/2)*exp(-x^2)
(※3991211251234741/4503599627370496*4^(1/2)/pi^(1/2)=1)

基因演算法

例：求 $f(x, y) = 20 + x^2 + y^2 - 10(\cos(2\pi x) + \cos(2\pi y))$ 的極小值

$$\text{限制式 } \begin{bmatrix} 1 & 1 \\ -12 & \\ 2 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \leq \begin{bmatrix} 2 \\ 2 \\ 3 \end{bmatrix}, \quad \begin{bmatrix} -5 \\ -5 \end{bmatrix} \leq \begin{bmatrix} x \\ y \end{bmatrix} \leq \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

function y = gatest(x)

y = 20 + x(1)^2 + x(2)^2 - 10 * (cos(2*pi*x(1))+cos(2*pi*x(2)));

```
>> A=[1 1;-1 2;2 1];
>> b=[2;2;3];
>> Lb=[-5;-5];
>> Ub=[3;4];
>> [x,fval,reason]=ga(@gatest,2,A,b,[],[],Lb,Ub)
```

執行結果：

x = 0.0059 0.0023

fval = 0.0080

reason = Optimization terminated: average change in the fitness value less than options.TolFun.

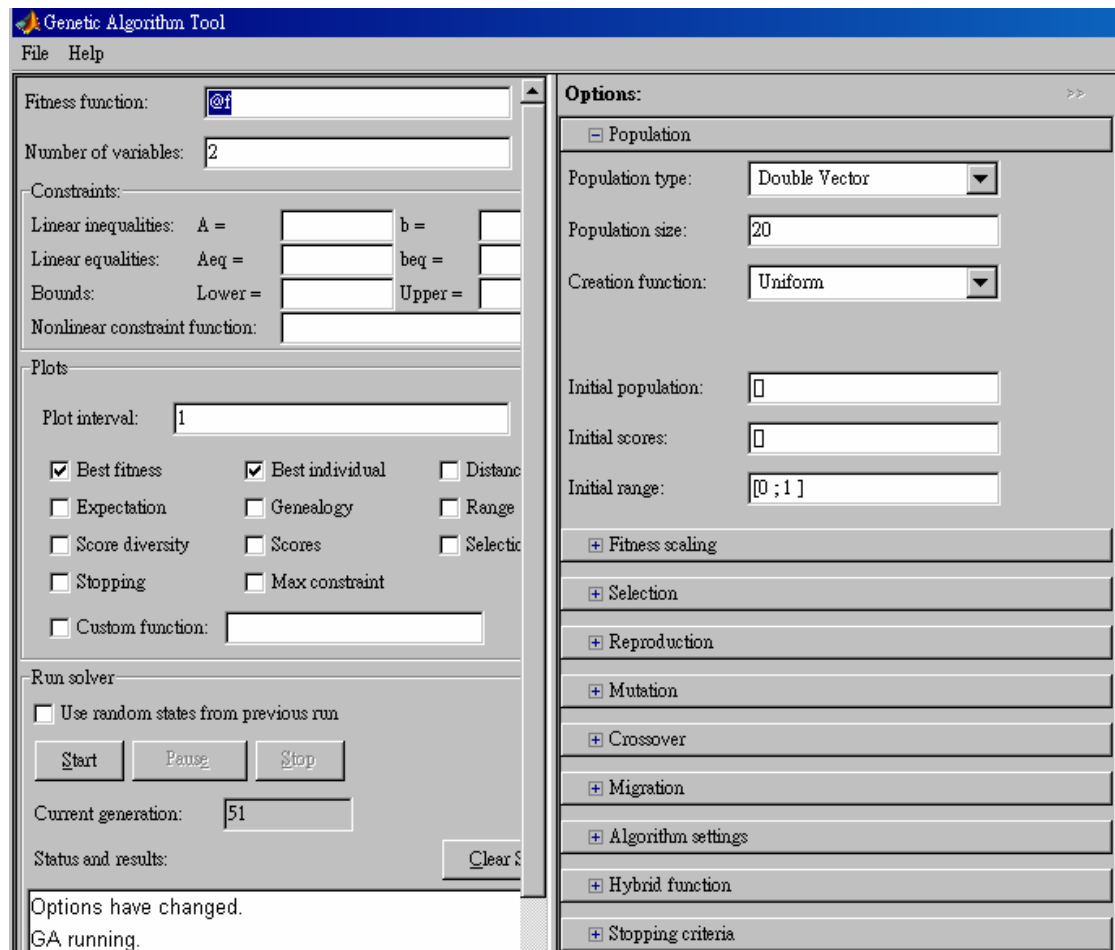
自訂函數

```
function v = f(x)
```

```
v = x(1).^2 + x(2).^2;
```

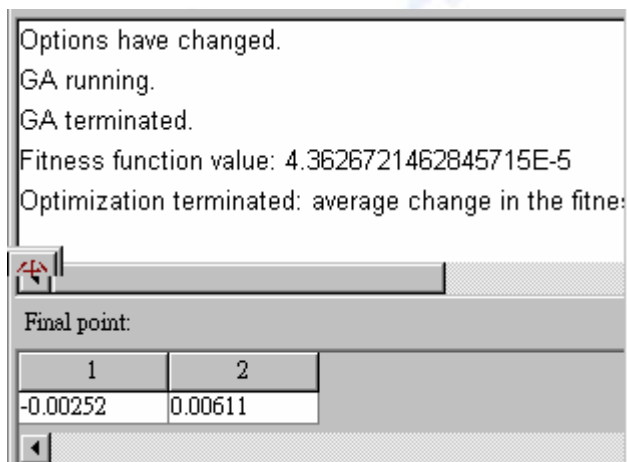
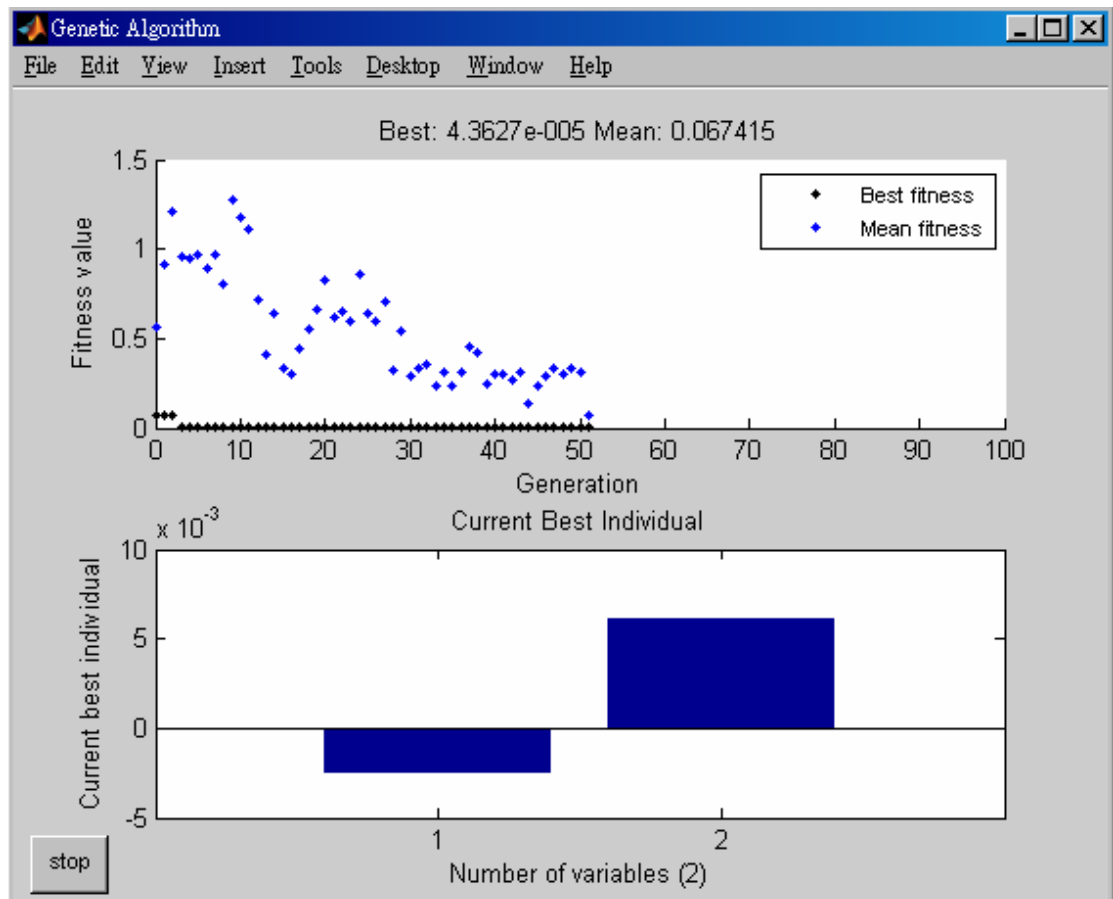
```
>> [x,fval,reason]=ga(@f,2)
```

或用 gatool 操作



執行結果：

林晶環老師



Optimization terminated: average change in the fitness value less than options.TolFun.

$x = 0.0044 \quad 0.0052$

$fval = 4.6427e-005$

reason = Optimization terminated: average change in the fitness value less than options.TolFun.

例：If $\frac{x^2}{25} + \frac{y^2}{9} = 1$, find the maximum value of $\sqrt{x^2 + (y-3)^2}$.