數值方法－使用MATLAB程式語言(第三版)

1 MATLAB 簡介

全華 Exit

1.1 MATLAB 套裝軟體

- MATLAB，其名稱來自「matrix laboratory」（矩陣實驗室）。
- MATLAB 在指令視窗內可以使用單一指令立即執行的模式或編輯一序列指令集成為指令碼(scripts) 的模式。

❖ MATLAB 提供給使用者下列優點：

1. 矩陣結構容易運算。
2. 大量強而有效的內建函數庫，目前仍持續增加發展中。
3. 靈巧有效的二維及立體繪圖設備。
4. 程式編寫系統方便使用者發展自己所需的軟體，甚至修改內建程式以供使用者自己特殊用途之需。
5. 函數集合成為工具箱，可以加入MATLAB 軟體核心。

❖ 有一些套裝軟體和MATLAB 類似：

- **APL**　這字母代表 A P rogramming L anguage
- **NAg 程式集(The NAg Library)**
- **Mathematica 和Maple**
- **其他套裝軟體**

1.2 MATLAB 的矩陣及其運算

- 矩陣是MATLAB 的基本，矩陣名稱必須以字母為首而後是數字與字母的組合，大小寫字母均可。
- 先開啟**command**（指令）視窗

```
>> A = [1 3 5;1 0 1;5 0 9]
```

- 分號(;) 代表著一行的結束並且另一行的開始。
- 當按下return 鍵後，矩陣顯示如下：

```
A =
     1     3     5
     1     0     1
     5     0     9
```


- 要執行一串連續的MATLAB 敘述，則利用MATLAB 的 **Editor**（編輯器）視窗將這些敘述輸入後建立指令碼，並以適當的命名以供後續使用。
- 僅有單一列或單一行的矩陣稱為向量(vector)。
- 列向量可寫成 $\mathbf{x}^T$ 。
- 矩陣乘法僅需鍵入 C=A*B 。

❖MATLAB 中一個非常有用的函數為 whos，這函數告訴我們當前工作環境中的內容

```
>> whos
  Name        Size                    Bytes  Class
  A           3x3                        72  double array
  B           3x3                        72  double array
  C           3x3                        72  double array

Grand total is 27 elements using 216 bytes
```



```
>> clear A
>> B = [ ];
>> C = zeros(4,4);

>> whos
  Name        Size                    Bytes  Class
  B           0x0                         0  double array
  C           4x4                       128  double array

Grand total is 16 elements using 128 bytes
```

❖ 容量的估計可以用 size 與 length 函數：

```
>> A = zeros(4,8);
>> B = ones(7,3);
>> [p q] = size(A)

p =
     4

q =
     8

>> length(A)

ans =
     8

>> L = length(B)

L =
     7
```


1.3 矩陣元素之運算操作

❖ 矩陣可以被個別或分區來運算

```
>> X(1,3) = C(4,5)+V(9,1)
>> A(1) = B(1)+D(1)
>> C(i,j+1) = D(i,j+1)+E(i,j)
```

❖ 小指令碼使用的方法，首先指定數據至一矩陣。

```
>> A = [2 3 4 5 6;-4 -5 -6 -7 -8; 3 5 7 9 1; ...
          4 6 8 10 12;-2 -3 -4 -5 -6]

A =
     2     3     4     5     6
    -4    -5    -6    -7    -8
     3     5     7     9     1
     4     6     8    10    12
    -2    -3    -4    -5    -6
```

❖ 注意刪節號... 用來表示MATLAB 描述接續著下一行，執行下列敘述

```
>> v = [1 3 5];
>> b = A(v,2)
```

```
b =
     3
     5
    -3
```


❖ 執行

```
>> C = A(v,:)
```

❖ 得到

```
C =
     2     3     4     5     6
     3     5     7     9     1
    -2    -3    -4    -5    -6
```

❖ 執行

```
>> D = zeros(3);
>> D(:,1) = A(v,2)
```

得到

```
D =
     3     0     0
     5     0     0
    -3     0     0
```


❖ 執行

```
>> E = A(1:2,4:5)
```

得到

```
E =
     5     6
    -7    -8
```

```
C1 = C;
C1(1:4:15) = 10

C1 =
    10     3     4     5    10
     3    10     7     9     1
    -2    -3    10    -5    -6
```

```
>> size(A)

ans =
     5     5

>> A(4,5)

ans =
    12
```

```
>> A(end-1,end)

ans =
    12
```


- reshape 函數可用來處理一個完整的矩陣，函數 reshape 可將一已知矩陣整形成另一指定大小的新矩陣。
- 例如，考慮矩陣 P 。

```
>> P = C(:,1:4)

P =
     2     3     4     5
     3     5     7     9
    -2    -3    -4    -5

>> reshape(P,6,2)

ans =
     2     4
     3     7
    -2    -4
     3     5
     5     9
    -3    -5
```



```
>> s = reshape(P,1,12);
>> s(1:10)

ans =
     2     3    -2     3     5    -3     4     7    -4     5
```

1.4 矩陣之轉置

❖轉置是將行列互換的一個簡單運算，在MATLAB內以符號 ' 來表示。

```
>> A = [1 2 3;4 5 6;7 8 9]

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

```
>> B = A'

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


❖使用 .' 也會的到相同的結果。若 A 是複數，則 MATLAB 運算子 ' 給出共軛複數轉置。

```
>>    =    +    +  ; +    +

A =
   1.0000 + 2.0000i   3.0000 + 5.0000i
   4.0000 + 2.0000i   3.0000 + 4.0000i

>> B = A'

B =
   1.0000 - 2.0000i   4.0000 - 2.0000i
   3.0000 - 5.0000i   3.0000 - 4.0000i
```

❖ 若要轉置而不共軛，則執行

```
>> C = A.'

C =
   1.0000 + 2.0000i   4.0000 + 2.0000i
   3.0000 + 5.0000i   3.0000 + 4.0000i
```


1.5 特殊矩陣

❖ 特定的矩陣：最常出現的是 ones(m,n)、zeros(m,n)、rand(m,n)、randn(m,n) 和 randi(p,m,n)。

❖ 函數 eye(m,n) 產生 *m* 列和 *n* 行的單位矩陣：

```
>> A = eye(3,4), B = eye(4,3)

A =
     1     0     0     0
     0     1     0     0
     0     0     1     0

B =
     1     0     0
     0     1     0
     0     0     1
     0     0     0
```


1.6 產生元素為指定值的矩陣

❖ linspace 也可以用來產生向量，使用者定義向量的起始值、終點值以及向量的元素數量。

```
>> w = linspace(-2,2,5)

w =
    -2    -1     0     1     2
```

```
>> w = linspace 0.2598,0.3024,5

w =
    0.2598    0.2704    0.2811    0.2918    0.3024
```

```
>> w = logspace 1,2,5

w =
   10.0000   17.7828   31.6228   56.2341  100.0000
```

❖ logspace 的使用者必須注意，若第二個參數是 pi 的話，值就會是 π 非10^{π} 。

```
>> w = logspace(1,pi,5)

w =
   10.0000    7.4866    5.6050    4.1963    3.1416
```

```
>> C = [2.3 4.9; 0.9 3.1];
>> D = [C ones(size(C)); eye(size(C)) zeros(size(C))]
```



```
D =
    2.3000    4.9000    1.0000    1.0000
    0.9000    3.1000    1.0000    1.0000
    1.0000         0         0         0
         0    1.0000         0         0
```

❖ MATLAB 函數 repmat 依照所需的次數複製給定矩陣

```
>> E = repmat(C,2,3)
```

```
E =
    2.3000    4.9000    2.3000    4.9000    2.3000    4.9000
    0.9000    3.1000    0.9000    3.1000    0.9000    3.1000
    2.3000    4.9000    2.3000    4.9000    2.3000    4.9000
    0.9000    3.1000    0.9000    3.1000    0.9000    3.1000
```

❖ MATLAB 函數 diag 允許從對角元素的特定向量來產生對角矩陣

```
>> H = diag([2 3 4])
```

產生

```
H =
     2     0     0
     0     3     0
     0     0     4
```


❖ 函數 diag 的第二個用法：

```
>> P = rand(3,4)

P =
    0.3825    0.9379    0.2935    0.8548
    0.4658    0.8146    0.2502    0.3160
    0.1030    0.0296    0.5830    0.6325
```

```
>> diag(P)

ans =
    0.3825
    0.8146
    0.5830
```

❖ 用MATLAB 函數blkdiag設定矩陣 A1 與 A2 如下：

```
>> A1 = [1 2 5;3 4 6;3 4 5];
>> A2 = [1.2 3.5,8;0.6 0.9,56];
```

```
>> blkdiag(A1,A2,78)

ans =
   1.0000    2.0000    5.0000         0         0         0         0
   3.0000    4.0000    6.0000         0         0         0         0
   3.0000    4.0000    5.0000         0         0         0         0
        0         0         0    1.2000    3.5000    8.0000         0
        0         0         0    0.6000    0.9000   56.0000         0
        0         0         0         0         0         0   78.0000
```


1.7 一些矩陣函數

❖ 若乘冪等於0.5，使用 sqrtm(A) 較好。

❖ 乘冪為−1 則使用 inv(A)。

❖ expm(A) ，計算矩陣A 的指數。

❖ MATLAB 函數 logm(A) 提供了以 e 為底數之 A 的主要對數。

❖ 例如

```
>> A = [61 45;60 76]

A =
    61    45
    60    76

>> B = sqrtm(A)

B =
    7.0000    3.0000
    4.0000    8.0000
```

```
>> B^2

ans =
   61.0000   45.0000
   60.0000   76.0000
```


1.8 以MATLAB 運算子\ 作矩陣除法

❖ 考慮矩陣方程式

$$\mathbf{Ax} = \mathbf{b} \qquad (1.1)$$

❖ 在MATLAB 內可以執行下列指令來求解 **x**

```
x = A\b
```

❖ 的MATLAB 的除法運算子\ 可用來解線性聯立方程式(1.1)。

1.9 元素逐一運算

- 元素逐一運算由運算子前放置一句點(.) 來達成。
- X.^Y 每個 X 的元素由 Y 與 X 相對應的元素來提升乘冪。
- 例子如下：

```
>> A = [1 2;3 4]

A =

     1     2
     3     4
```



```
>> B = [5 6;7 8]

B =

     5     6
     7     8
```

- 一般矩陣乘法：

```
>> A*B

ans =

    19    22
    43    50
```

❖(.) 運算元可得

```
>> A.*B

ans =
     5    12
    21    32
```

```
>> A.^B

ans =
           1          64
        2187       65536
```


❖元素逐一運算其中一個重要的運用是繪圖。例如

```
>> x = -1:0.1:1;
>> y = x.*cos(x);
>> y1 = x.^3.*(x.^2+3*x+sin(x));
```

1.10 純量運算與函數

- A 可以表示一個純量或者是矩陣，在運算的程序才看到差異。
- 例如

```
>> x = 2;
>> y = x^2+3*x-7

y =
     3

>> x = [1 2;3 4]
```



```
x =
     1     2
     3     4

>> y = x.^2+3*x-7

y =
    -3     3
    11    21
```

```
>> y = x^2+3*x-7

y =
     3     9
    17    27
```

❖MATLAB 內建有非常大量的數學函數。

❖函數的例子：

```
>> x = [-4 3];
>> abs(x)

ans =
     4     3

>> x = 3+4i;
>> abs(x)
```


表1.1　MATLAB 數學函數

函數	函數提供	範例
`sqrt(x)`	square root of x	`y = sqrt(x+2.5);`
`abs(x)`	if x is real, gives positive value of x	
	If x is complex, gives scalar measure of x	`d = abs(x)*y;`
`real(x)`	real part of x when x is complex	`d = real(x)*y;`
`imag(x)`	imaginary part of x when x is complex	`d = imag(x)*y;`
`conj(x)`	complex conjugate of x	`x = conj(y);`
`sin(x)`	sine of x in radians	`t = x+sin(x);`
`asin(x)`	inverse sine of x returned in radians	`t = x+sin(x);`
`sind(x)`	sine of x in degrees	`t = x+sind(x);`
`log(x)`	log to base e of x	`z = log(1+x);`
`log10(x)`	log to base 10 of x	`z = log10(1-2*x);`
`cosh(x)`	hyperbolic cosine of x	`u = cosh(pi*x);`
`exp(x)`	exponential of x, i.e., e^x	`p = .7*exp(x);`
`gamma(x)`	gamma function of x	`f = gamma(y);`
`bessel(n,x)`	nth-order Bessel function of x	`f = bessel(2,y);`

```
ans =
     5

>> imag(x)

ans =
     4

>> y = sin(pi/4)

y =
    0.7071

>> x = linspace(0,pi,5)
```



```
x =
         0    0.7854    1.5708    2.3562    3.1416

>> sin(x)

ans =
         0    0.7071    1.0000    0.7071    0.0000

>> x = [0 pi/2;pi 3*pi/2]

x =
         0    1.5708
    3.1416    4.7124

>> y = sin(x)

y =
         0    1.0000
    0.0000   -1.0000
```

❖ 一些函數提供了特別的運算給一些重要或是一般的數值程序，這些函數通常需要一個以上的輸入參數並且提供多個輸出。例如， bessel(n, x) 給出 x 的第 n階貝所函數；敘述 y = fzero(fun, x0) 決定了函數 fun 接近 x0 的根，其中 fun是個由使用者定義的函數，讓我們找出方程式的根。

❖ 下列的指令碼是使用計時函數，用來估計執行 1000×1000 線性聯立方成組的執行時間。

```
% e3s107.m  Solves a 1000x1000 linear equation system
A = rand(1000); b = rand(1000,1);
T_before = clock;
tic
t0 = cputime;
y = A\b;
timetaken = etime(clock,T_before);
tend = toc;
t1 = cputime-t0;
disp('etime      tic-toc    cputime')
fprintf('%5.2f %10.2f %10.2f\n\n', timetaken,tend,t1);
```

```
etime      tic-toc    cputime
 0.30        0.31       0.30
```

1.11 字串變數

❖ 字串可用特殊的MATLAB 字串函數來處理。

❖ 範例：

```
>> s1 = 'Matlab ', s2 = 'is ', s3 = 'useful'

s1 =
Matlab

s2 =
is
```

```
s3 =
useful
```


❖ 我們可以串接字串，如下使用用方括弧：

```
>> sc = [s1 s2 s3]

sc =
Matlab is useful
```

```
>> sc(2)

ans =
a
```

```
>> sc(3:10)

ans =
tlab is
```

❖ 藉由轉置字串向量將字串以垂直方式顯示

```
>> sc(1:3)'

ans =
M
a
t
```

❖ 也可以將字串的子集合順序反向並指定至另一個字串，如下；

```
>>a = sc(6:-1:1)

a =
baltaM
```


❖ 沿用先前所定義的字串sc 可以得到

```
>> sd = 'Numerical method'
>> s = [sc; sd]

s =
Matlab is useful
Numerical method
```

❖ 用strrep 將字串進行替換，如下：

```
>> strrep(sc,'useful','super')

ans =
Matlab is super
```

❖ 用 findstr 來找出在某一個字串中特定的字元或是字串

```
>> findstr(sd,'e')

ans =
     4    12
```

```
>> findstr(sd, 'meth')

ans =
    11
```

```
>> findstr(sd,'E')

ans =
     [ ]
```


❖ 字串可以利用函數 double 或者是執行任何數學運算而轉換成相對應的ASCII碼。

```
>> p = double(sd(1:9))

p =
    78   117   109   101   114   105    99    97   108

>> q = 1*sd(1:9)

q =
    78   117   109   101   114   105    99    97   108
```

❖ 使用MATLAB 的 char 函數可以將ASCII 碼的向量轉換成字串。

❖ 例如：

```
>> char(q)

ans =
Numerical
```

```
>> char(q+3)

ans =
Qxphulfdo
```



```
>> char((q+3)/2)

ans =
(<84:6327

>> double(ans)

ans =
    40    60    56    52    58    54    51    50    55
```

❖字串 '123' 非常重要且並非與數值123 相同，因此

```
>> a = 123

a =
   123

>> s1 = '123'

s1 =
123
```


❖whos 可以顯示出變數 a 和 s1 的類別如下：

```
>> whos
  Name        Size                     Bytes  Class
  a           1x1                          8  double array
  s1          1x3                          6  char array

Grand total is 4 elements using 14 bytes
```

❖可以用函數str2num 與 str2double 轉換字串至相對應的數值，如下：

```
>> x=str2num('123.56')

x =
   123.5600
```

```
>> x = str2num('1+2j')

x =
1.0 + 2.0000i
```

```
>> x = str2num('1+2 j')

x =
   3.0000                    0 + 1.0000i
```

```
>> x = str2double('1+2 j')

x =
1.0 + 2.0000i
```


- bin2dec('111001') 或 bin2dec('111 001') 回傳57
- dec2bin(57) 回傳字串 '111001'
- int2str([3.9 6.2]) 回傳字串 '4 6'
- num2str([3.9 6.2]) 回傳字串 '3.9 6.2'
- str2num('3.9 6.2') 回傳 3.9000 6.2000
- strcat('how ','why ','when') 回傳字串 'howwhywhen'
- strcmp('whitehouse','whitepaint') 回傳0 因為字串不相同

- strncmp('whitehouse','whitepaint',5)回傳 1 因為字串的第 5 個字元是相同的
- date 回傳現在日期，以 24-Aug-2011 此格式

1.12 MATLAB 之輸入及輸出

❖ disp 函數將文字及數值顯示在螢幕上。

```
>> disp('This will display this test')
This will display this test
```

```
>> x = 2.678;
>> disp(['Value of iterate is ', num2str(x), ' at this stage'])
```

```
Value of iterate is 2.678 at this stage
```

❖ fprintf 是更具彈性的函數

```
fprintf('filename','format_string',list);
```

❖ 可能使用的格式字串元素是：

- %P.Qe 設定指數的格式
- %P.Qf 設定定點的格式
- %P.Qg 變成 %P.Qe 或 %P.Qf 視何者較短而定
- \n 設定新一列


```
>> x = 1007.461; y = 2.1278; k = 17;
>> fprintf('\n x = %8.2f y = %8.6f k = %2.0f \n',x,y,k)
```

```
x =  1007.46 y = 2.127800 k = 17
```

```
>> p = sprintf('\n x = %8.2f y = %8.6f k = %2.0f \n',x,y,k)
```

```
p =

 x =  1007.46 y = 2.127800 k = 17
```

❖ 互動式的方法來輸入是使用函數input，形式如下

```
>> variable = input('Enter data: ');
Enter data: 67.3
```

❖ 函數 input 的另一個形式是允許字串輸入。

```
>> variable = input('Enter text: ','s');
Enter text: Male
```

❖ 函數 load 可以由磁碟載入檔案，使用

```
load filename
```

```
>> load sunspot.dat
```



```
>> x = 1:5; y = sin(x); z = cos(x);
>> whos
  Name        Size            Bytes  Class
  x           1x5                40  double array
  y           1x5                40  double array
  z           1x5                40  double array
>> save test001
>> clear all, whos       Nothing listed
>> load test001
>> whos
  Name        Size            Bytes  Class
  x           1x5                40  double array
  y           1x5                40  double array
  z           1x5                40  double array
>> x = 1:5; y = sin(x); z = cos(x);
```

```
>> save test002 x y
>> clear all,  whos  Nothing listed
>> load test002 x y, whos
  Name       Size            Bytes  Class
  x          1x5                40  double array
  y          1x5                40  double array
```

```
>> clear all, whos   Nothing listed
>> load test002 x, whos
  Name       Size            Bytes  Class
  x          1x5                40  double array
```


❖ 以向量 g 來呼叫：

```
>> p = 1:6;
>> whos
  Name       Size            Bytes  Class
  p          1x6                48  double array

>> csvwrite('test003',p)
>> clear
>> g = csvread('test003')
g =
     1     2     3     4     5
```

1.13 MATLAB 繪圖

- plot(x, y) 繪出向量 x 對 y 的圖，如果 x 和 y 都是矩陣則繪 x 的第1 行對 y 的第1 行的圖，然後逐對 x 及 y 重覆。
- plot(x1, y1, 'type1', x2, y2, 'type2') 使用 type1 的線或點型繪出 x1對 y1 的圖，然後使用 type2 的線或點型繪出 x2 對 y2 的圖。

表1.2 繪圖所使用的符號及字元

線	符號	點	符號	顏色	文字
solid	-	point	.	yellow	y
dashed	--	plus	+	red	r
dotted	:	star	*	green	g
dashdot	-.	circle	o	blue	b
		x mark	×	black	k

- title('title') 顯示在引號內的標題於圖形上方。
- xlabel('x_axis_name') 在軸下顯示引號內的名稱。
- ylabel('y_axis_name') 在軸左側顯示引號內的名稱。
- grid 在圖形套上格狀網。
- text(x,y, 'text-at-x,y') 在圖形視窗中的位置(x , y) 處顯示文字資訊，x 及 y 的單位是以目前繪圖軸的單位量取的。文字可能顯示在一點或多點，視 x 和 y 是否為向量而定。

- gte xt('text') ，由滑鼠選定位置置放所要的文字，然後再按下滑鼠按鈕。
- ginput 允許從圖形視窗擷取資訊。

❖ginput 有兩種主要形式

```
[x,y] = ginput
```

例如：

```
[x,y] = ginput(n)
```

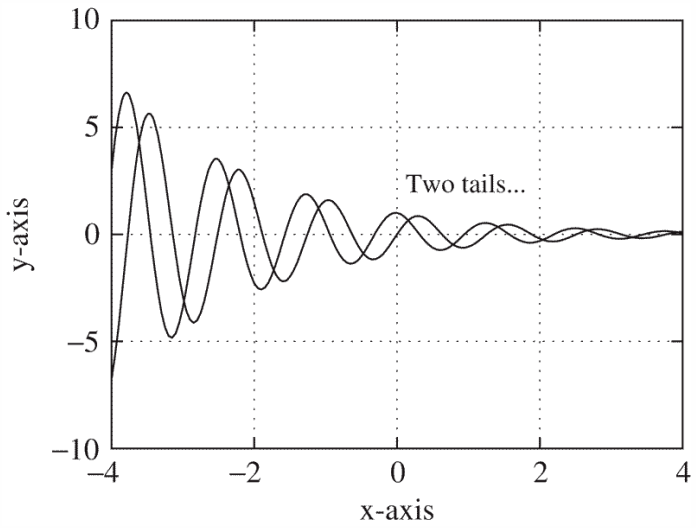


圖1.1　使用plot(x, y) 及hold 敘述得到的圖形重疊 。

❖ 函數 hold 是用來保證兩個圖形重疊。

```
% e3s101.m
x = -4:0.05:4;
y = exp(-0.5*x).*sin(5*x);
figure(1), plot(x,y)
xlabel('x-axis'), ylabel('y-axis')
hold on
y = exp(-0.5*x).*cos(5*x);
plot(x,y), grid
gtext('Two tails...')
hold off
```

❖ 函數 fplot 允許使用者在給定的界限內繪出先前已定義的函數。

```
% e3s102.m
y = @(x) sin(x.^3);
x = 2:.04:4;
figure(1)
plot(x,y(x),'o-')
xlabel('x'), ylabel('y')
figure(2)
fplot(y,[2 4])
xlabel('x'), ylabel('y')
```


❖ 注意 figure(1) 及 figure(2) 將輸出繪至不同視窗。

❖ ezplot 允許參數繪圖與三維繪圖，例如

```
>> ezplot(@(t) (cos(3*t)), @(t) (sin(1.6*t)), [0 50])
```

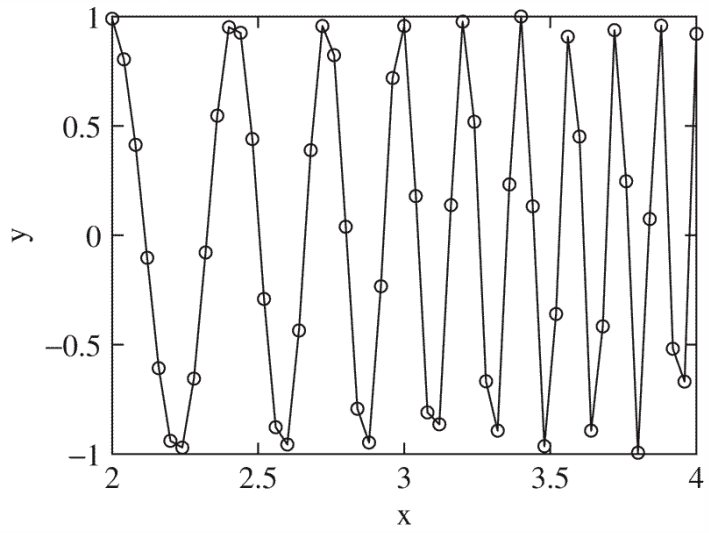


圖1.2　使用51 個等間隔的繪圖點來畫出 y =sin(x^3) 。

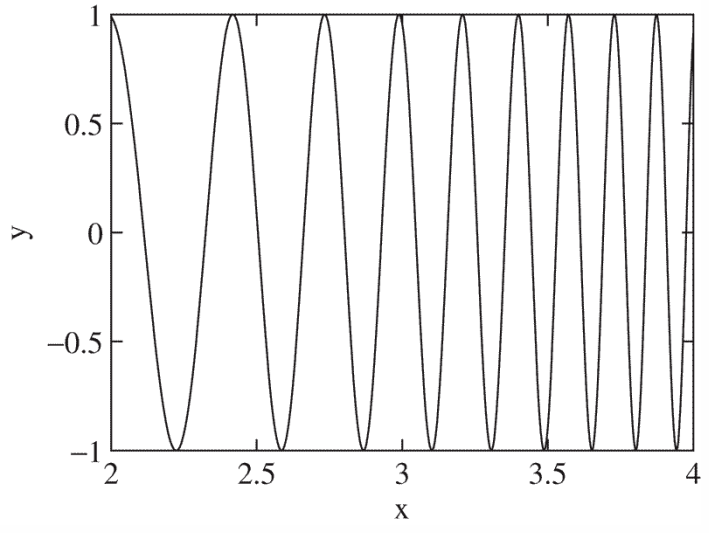


圖1.3　使用fplot 函數選定合適的點數來繪出y =sin(x^3) 。

```
>> x = -4:0.0011:4;
>> y =1./(((x+2.5).^2).*((x-3.5).^2))+1./((x-1).^2);
>> plot(x,y)
>> ylim([0,10])
```

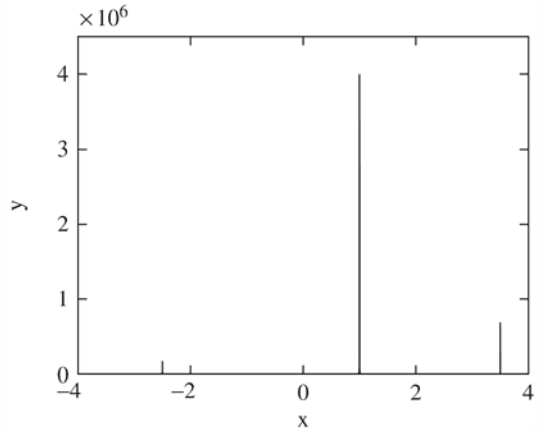


圖1.4　函數於範圍−4 至4 之間所繪的圖，最大值為4×10^6 。

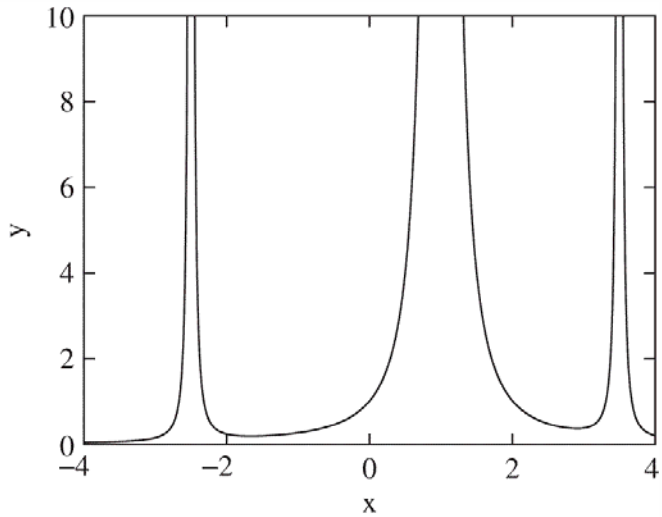


圖1.5　與圖1.4 相同的函數所繪出的圖，但y 軸的範圍有所限制。

❖ 函數hold on 打開保留裝置而 hold off 則將之關閉。圖形視窗可用 clf 清除。

```
% e3s103.m
x = 0.1:.1:5;
subplot(2,3,1), plot(x,x)
title('plot of x'), xlabel('x'), ylabel('y')
subplot(2,3,2), plot(x,x.^2)
title('plot of x^2'), xlabel('x'), ylabel('y')
subplot(2,3,3), plot(x,x.^3)
title('plot of x^3'), xlabel('x'), ylabel('y')
subplot(2,3,4), plot(x,cos(x))
title('plot of cos(x)'), xlabel('x'), ylabel('y')
subplot(2,3,5), plot(x,cos(2*x))
title('plot of cos(2x)'), xlabel('x'), ylabel('y')
subplot(2,3,6), plot(x,cos(3*x))
title('plot of cos(3x)'), xlabel('x'), ylabel('y')
```

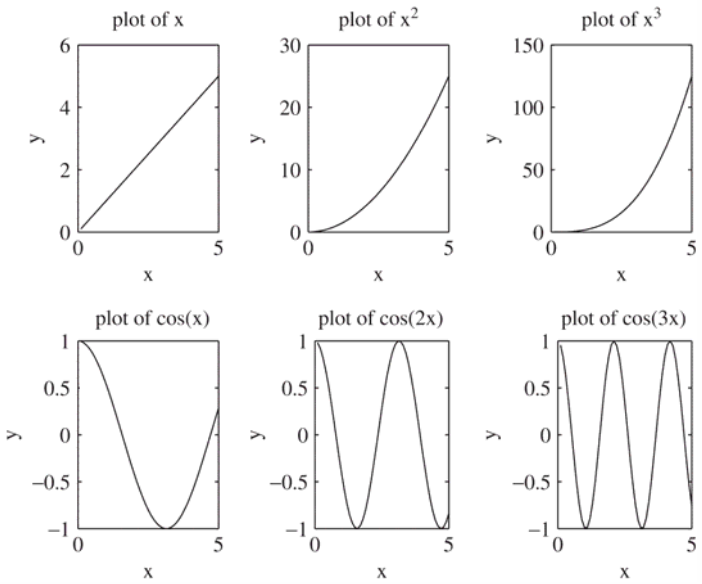



圖1.6　函數subplot 的使用範例。

```
>> p=roots([1 0 0 0 0 1])

p =
  -1.0000
  -0.3090 + 0.9511i
  -0.3090 - 0.9511i
   0.8090 + 0.5878i
   0.8090 - 0.5878i

>> pm = abs(p.')

pm =
    1.0000    1.0000    1.0000    1.0000    1.0000

>> pa = angle(p.')

pa =
    3.1416    1.8850   -1.8850    0.6283   -0.6283

>> subplot(1,2,1), polar(pa,pm,'ok')
>> subplot(1,2,2), compass(real(p),imag(p),'k')
```

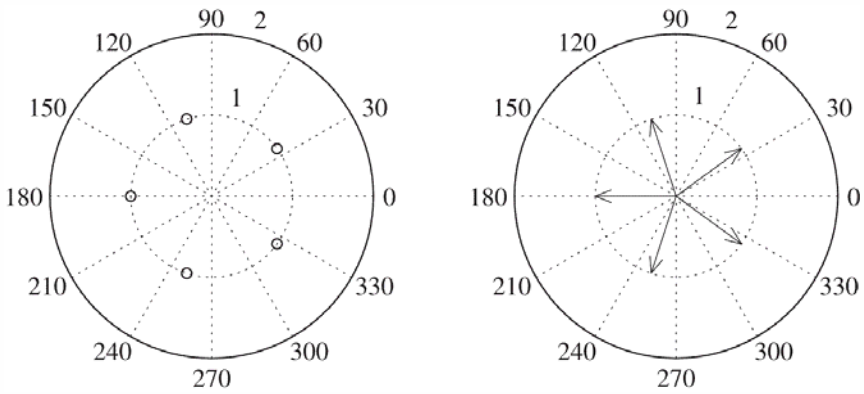



圖1.7　polar 和 compass 繪出 $x^5-1=0$ 的根 。

1.14 立體繪圖

- MATLAB 提供使用者方便有效力的繪製立體圖形。
- 這些函數是meshgrid、mesh、surf 1、 contour和 contour3。
- 繪出函數

$$z=(-20x^2+x)/2+(-15y^2+5y)/2 \qquad x=-4:0.2:4 \quad \text{及} \quad y=-4:0.2:4$$


```
% e3s105.m
[x,y] = meshgrid(-4.0:0.2:4.0,-4.0:0.2:4.0);
z = 0.5*(-20*x.^2+x)+0.5*(-15*y.^2+5*y);
figure(1)
surfl(x,y,z); axis([-4 4 -4 4 -400 0])
xlabel('x-axis'), ylabel('y-axis'), zlabel('z-axis')
figure(2)
contour3(x,y,z,15); axis([-4 4 -4 4 -400 0])
xlabel('x-axis'), ylabel('y-axis'), zlabel('z-axis')
figure(3)
contourf(x,y,z,10)
xlabel('x-axis')   label(' -axis')
```

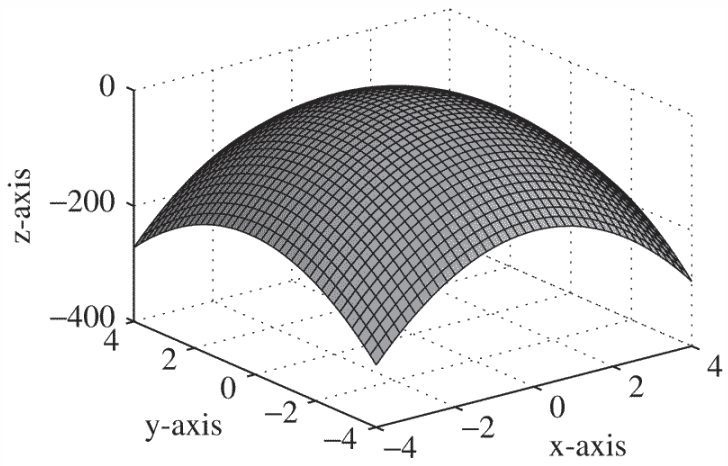


圖1.8　使用預設觀測角的立體圖形。

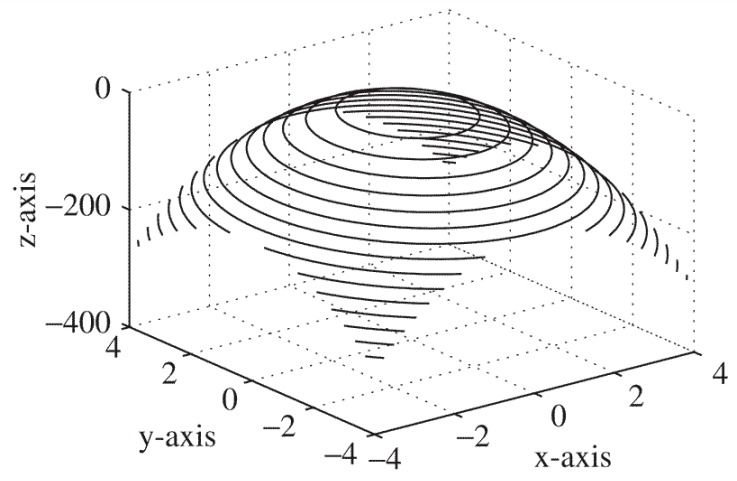


圖1.9　立體輪廓圖線繪製。

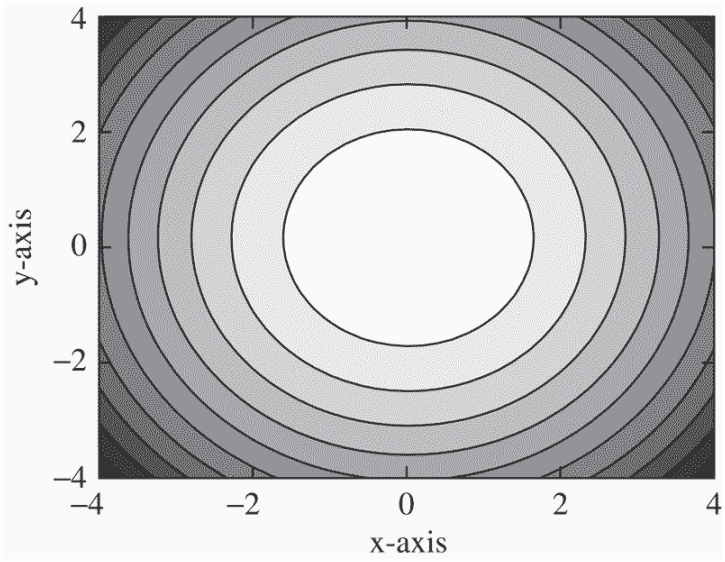


圖1.10　填色的輪廓繪圖。

1.15　圖形操作- 握把式圖形

❖ 握把式圖形(Handle Graphics) 讓使用者選擇字型、線條粗細、符號型態及1.15　圖形操作- 握把式圖形1-33大小、軸型態以及其他對於繪圖的一些特徵。

```
>> x = -4:.1:4;
>> y = cos(x);
>> h = plot(x,y);
>> h1 = title('cos graph')
```

```
>> get(h)
                    Color: [0 0 1]
                EraseMode: 'normal'
                LineStyle: '-'
                LineWidth: 0.5000
                   Marker: 'none'
               MarkerSize: 6
          MarkerEdgeColor: 'auto'
          ..........................[etc]
```



```
>> get(h1)
FontName = Helvetica
FontSize = [10]
FontUnits = points
  HorizontalAlignment = center
LineStyle = -
LineWidth = [0.5]
Margin = [2]
Position = [-0.00921659 1.03801 1.00011]
Rotation = [0]
String = cos graph
..........................[etc]
```

```
% e3s121.m
% Example for Handle Graphics
x = -5:0.1:5;
subplot(1,3,1)
e1 = plot(x,sin(x)); title('sin x')
```

```
subplot(1,3,2)
e2 = plot(x,sin(2*round(x))); title('sin round x')
subplot(1,3,3)
e3 = plot(x,sin(sin(5*x))); title('sin sin 5x')
```

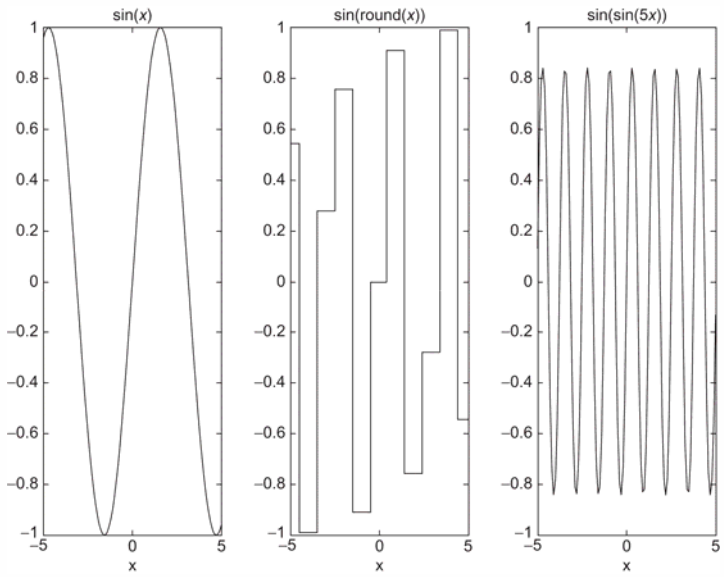



圖1.11　握把圖形的圖形解說方面 。

❖指令碼使用一連串的 set 敘述修改後如下：

```
% e3s122.m
% Example for Handle Graphics
x = -5:0.1:5;
s1 = subplot(1,3,1);
e1 = plot(x,sin(x)); t1 = title('sin(x)');
s2 = subplot(1,3,2);
e2 = plot(x,sin(2*round(x))); t2 = title('sin(round(x))');
s3 = subplot(1,3,3);
e3 = plot(x,sin(sin(5*x))); t3 = title('sin(sin(5x))');
% change dimensions of first subplot
```



```
set(s1,'Position',[0.1 0.1 0.2 0.5]);
%change thickness of line of first graph
set(e1,'LineWidth',6)
set(s1,'XTick',[-5 -2  0 2  5])
%Change all titles to italics
set(t1,'FontAngle','italic'), set(t1,'FontWeight','bold')
set(t1,'FontSize',16)
set(t2,'FontAngle','italic')
set(t3,'FontAngle','italic')
%change dimensions of last subplot
set(s3,'Position',[0.7 0.1 0.2 0.5]);
```

❖ Position 敘述有下列的值

```
[shift from left, shift from bottom, width, height]
```

```
set(s3,'Position',[0.7 0.1 0.2 0.5]);
```

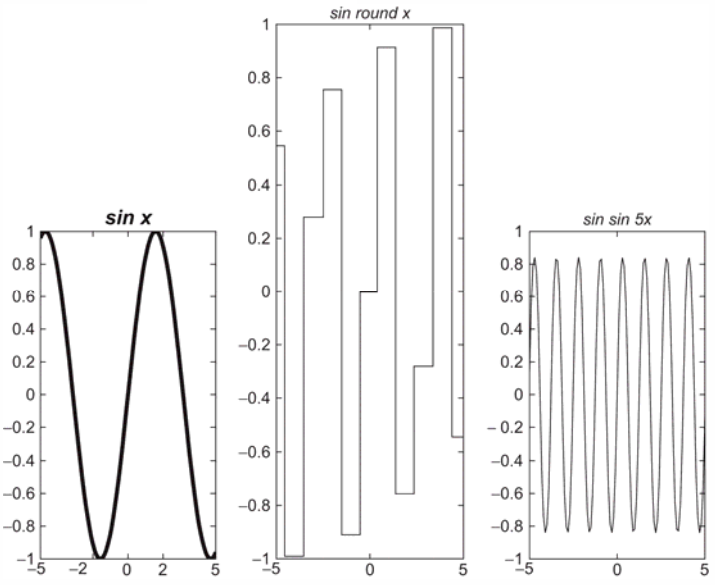



圖1.12　在圖1.11 中所介紹的握把圖形特徵的函數繪出的圖 。

❖使用 gca 在改變軸的這種屬性：

```
>> x = -1:0.1:2; h = plot(x,cos(2*x));
```

```
>> get(gca,'FontWeight')

ans =
normal

>> set(gca,'FontWeight','bold')
>> set(gca,'FontSize',16)
>> set(gca,'XTick',[-1 0 1 2])
```

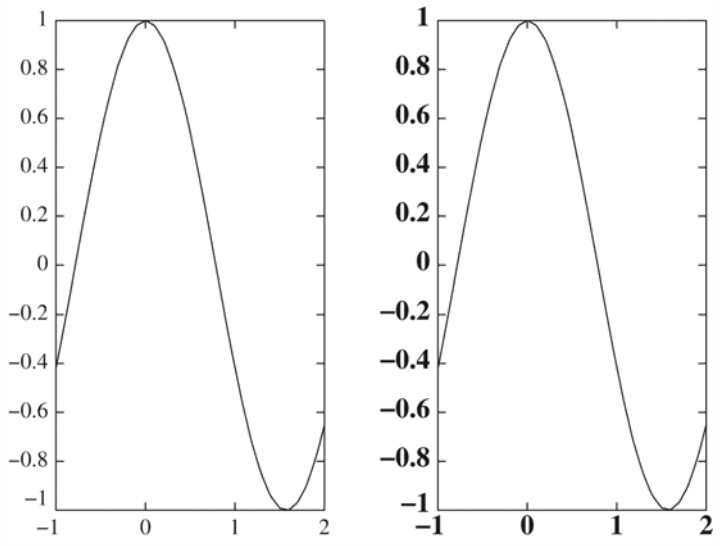



圖1.13　cos(2x) 的圖形，右手邊的圖是經過握把圖形加強 。

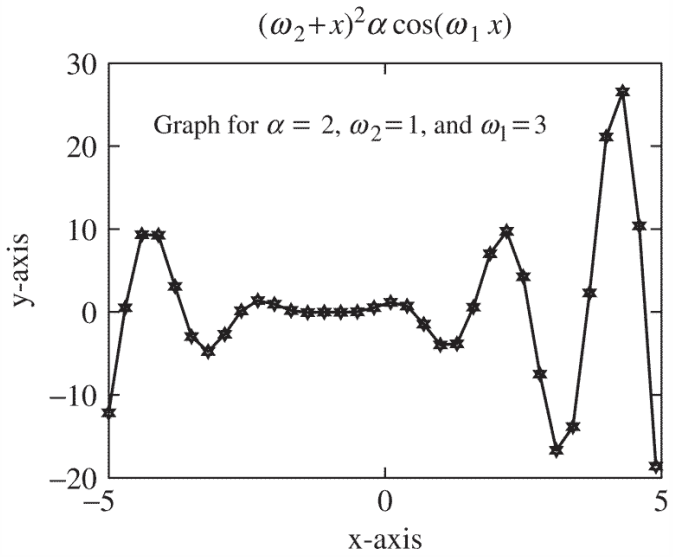


圖1.14　$(\omega_2+x)^2\alpha\cos(\omega_1 x)$的圖 。


```
% e3s104.m
% Example of the use of special graphics parameters in MATLAB
% illustrates the use of superscripts, subscripts,
% fontsize and special characters
x = -5:.3:5;
plot(x,(1+x).^2.*cos(3*x),...
'linewidth',1,'marker','hexagram','markersize',12)
title('(\omega_2+x)^2\alpha cos(\omega_1x)','fontsize',14)
xlabel('x-axis'), ylabel('y-axis','rotation',0)
gtext('graph for \alpha = 2,\omega_2 = 1, and \omega_1 = 3')
```

❖利用所介紹過的反斜線「\」來使用希臘字體的延伸符號

- alpha 給出α
- beta 給出β
- gamma 給出γ

❖例如：

```
title('(\omega_2+x)^2\alpha*cos(\omega_1*x)','fontsize',14)
```

$(\omega_2 + x)^2 \alpha * \cos(\omega_1 * x)$

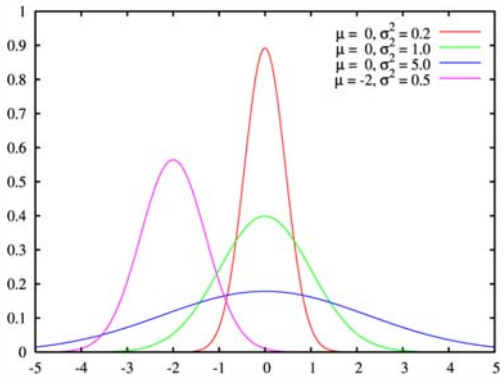
❖ 下列敘述增加在所有產生圖形輸出的指令碼

```
set(0,'defaultaxesfontsize',16)
set(0,'defaultaxesfontname','Times New Roman')
set(0,'defaulttextfontsize',12)
set(0,'defaulttextfontname','Times New Roman')
axes('position',[0.30 0.30 0.50 0.50])
```


練習: 常態分佈 (Normal distribution, Gaussian distribution)

❖ 常態分佈的機率密度函數均值為 μ ，變異數為 σ^2 (或標準差 σ)

$$f(x;\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$



練習: 常態分佈 MATLAB程式

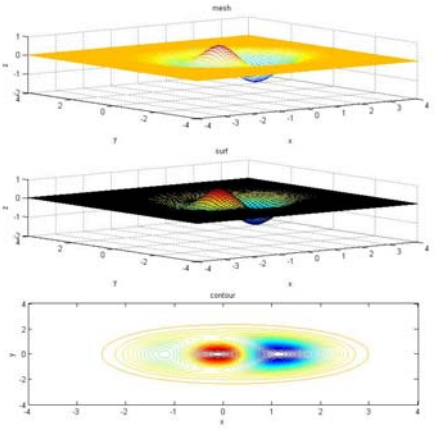
```
x=-5:0.01:5;
u=0;
d1=sqrt(0.2);
f1=(1/(d1*sqrt(2*pi)))*exp(-((x-u).^2)/(2*d1.^2));
figure(1)
plot(x,f1,'r')
hold on
d2=sqrt(1);
f2=(1/(d2*sqrt(2*pi)))*exp(-((x-u).^2)/(2*d2.^2));
plot(x,f2,'g')
d3=sqrt(5);
f3=(1/(d3*sqrt(2*pi)))*exp(-((x-u).^2)/(2*d3.^2));
plot(x,f3,'b')
u1=-2;
d4=sqrt(0.5);
f4=(1/(d4*sqrt(2*pi)))*exp(-((x-u1).^2)/(2*d4.^2));
plot(x,f4,'b')
hold off
```


練習: 三維繪圖

❖ 在x=-4:0.1:4與y=-4:0.1:4範圍中，使用mesh、contour、surf繪圖，函數為：

$$z = (1-x^2)e^{-p} - pe^{-p} - e^{-(x+1)^2-y^2}$$

$$p = x^2 + y^2$$



練習: 三維繪圖

```
x=-4:0.1:4; y=-4:0.1:4;
[x,y]=meshgrid(x,y);
p=x.^2+y.^2;
z=(1-x.^2).*exp(-p)-p.*exp(-p)-exp(-(-x+1).^2-y.^2);
subplot(3,1,1)
mesh(x,y,z)
xlabel('x'), ylabel('y'), zlabel('z'), title('mesh')
subplot(3,1,2)
surf(x,y,z)
xlabel('x'), ylabel('y'), zlabel('z'), title('surf')
subplot(3,1,3)
contour(x,y,z,50)
xlabel('x'), ylabel('y'), zlabel('z'), title('contour')
```


1.16 在MATLAB內編寫程式

- 程式是在**edit**（編輯）視窗內編寫而不是在**command**視窗中編輯，command視窗只允許一次一個指令或同一行內的數個指令。
- 符號 % 之後的任何文字皆視為註解。另外有一些變數名稱已先定義其數值以方便使用者。

- pi　　等同於π
- inf　　除以0之後的結果
- eps　　設定成特別的機器精確度
- realmax　　最大正浮點數
- realmin　　最小正浮點數
- NaN　　「Not-a-Number」，由零除以零產生的非數目值
- i,j　　兩者皆為$\sqrt{-1}$

❖ 在MATLAB 程式內，敘述的指定型式為

```
variable = <expression>;
```

- ^　乘冪
- *　乘號
- / 除號
- +　加號
- - 減號

```
% e3s106.m
% Matrix calculations for two matrices A and B
A = [1 2 3;4 5 6;7 8 9];
B = [5 -6 -9;1 1 0;24 1 0];
% Addition. Result assigned to C
C = A+B; disp(C)
% Multiplication. Result assigned to D
D = A*B; disp(D)
% Division. Result assigned to E
E = A\B; disp(E)
```

❖ 為了能重覆執行某些敘述， for 迴圈可以使用，形式如下：

```
for <loop_variable> = <loop_expression>
    <statements>
end
```


for 迴圈

❖ 為了能重複執行某些程式

```
for <loop_variable> = <loop_expression>
    <statements>
end
```

❖ 例如

```
for i = 1:4
    d(i) = i^3;
end
```

可得d = 1 8 27 64

```
p = 1;
prime_numbs = [2 13 5 11 7 3];
for a = prime_numbs
    p = p*a;
end
p
```


練習: for 迴圈

產生一個6x6的矩陣h，其中為於第i列第j行的元素為

$$h_{i,j} = \frac{1}{i + j - 1}$$

練習: for 迴圈

```
h=zeros(6);
for i=1:6
   for j=1:6
      h(i,j)=1/(i+j-1);
   end
end
disp(h)
```


練習: for 迴圈

❖ 找出n! > 10^{100}值，使用break

練習: for 迴圈

```
for i=1:1000
   if prod(1:i)>1e100
      fprintf('%g!=%e>1e100\n',i,prod(1:i));
      break;
   end
end
```


while迴圈

❖ 為了滿足某些限制條件而執行某些程式

```
while <while_expression>
      <statements>
end
```

❖ <while_expression> 是一關係表示式，其型式為 e1。e2 ，其中 e1 和 e2 是一般的代表式，。是如下定義的關係運算元：

- == 等號
- <= 小於或等於
- >= 大於或等於
- ~= 不等於
- < 小於
- > 大於

❖ 下列邏輯運算元用來結合表示式子的關係

- & 和(and) 運算元
- | 或許(or) 運算元
- ~ 非(not) 運算元
- && 純量和(and) 運算元（若第一個條件為假則第二個不評估）
- || 或許(or) 運算元（若第一個條件為真則第二個不評估）

❖while 迴圈的例子

```
dif = 1;
x2 = 1;
while dif>0.0005
    x1 = x2-cos(x2)/(1+x2);
    dif = abs(x2-x1);
    x2 = x1;
end

x = [1 2 3];
y = [4 5 8];
while sum(x) ~= max(y)
    x = x.^2;
    y = y+x;
end
```


練習：while迴圈

❖使用while指令找出最小的n值，使得n! > 10^{100}

練習：while迴圈

```
n=1;
while prod(1:n)<1e100
   n=n+1;
end
fprintf('%g!=%e>1e100\n',n,prod(1:n));
```


if 敘述

❖ 程式內的指令執行時，為了改變執行的順序

```
if < if_expression1>
    <statements>
elseif < if_expression2>
    <statements>
elseif < if_expression3>
    <statements>
...
...
else
    <statements>
end
```

❖ 關係運算元可以由邏輯運算元組合。例如：

```
for k = 1:n
    for p = 1:m
        if k == p
            z(k,p) = 1;
            total = total+z(k,p);
        elseif k<p
            z(k,p) = -1;
            total = total+z(k,p);
        else
            z(k,p) = 0;
        end
    end
end

if (x~=0) & (x<y)
   b = sqrt(y-x)/x;
   disp(b)
end
```


練習：if敘述

❖ 根據向量y的元素值為奇數或偶數，來顯示不同的訊息

❖ y=[0 3 4 1 6];

練習: if敘述

```
y=[0 3 4 1 6];
for i=1:length(y)
    if rem(y(i),2)==0
        fprintf('y(%g)=%g is even.\n',i,y(i));
    else
        fprintf('y(%g)=%g is odd.\n',i,y(i));
    end
end
```


練習: for迴圈 加 if敘述

❖ 已知n個元素的向量x，寫一個matlab程式來得乘積為

$$p_k = x_1 x_2 \ldots x_{k-1} x_{k+1} \ldots x_n$$

其中n=6, x_1=1, x_2=2,…,x_6=6, k=1,2,…,n

```
Vector of data

x =

     1     2     3     4     5     6

Vector of products p_k

p =

   720   360   240   180   144   120
```

練習: for迴圈

```
clear all
close all
n = 6;
disp('Vector of data')
x = 1:n
for j = 1:n
    p(j) = 1;
    for i = 1:n
        if i~=j
            p(j) = p(j)*x(i);
        end
    end
end
disp('Vector of products p_k')
p
```


switch敘述

❖ 是另一種 if 的結構，特別是在有很多選項考慮的時候使用

```
switch <condition>
    case
        statements
    case ref2
        statements
    case ref3
        statements
    otherwise
        statements
end
```

```
 x = 1:.01:10; n = 2
switch n
    case 1
        plot(x,log(x));
    case 2
        plot(x,x.*log(x));
    case 3
        plot(x,x./(1+log(x)));
    otherwise
        disp('That was an invalid selection.')
end
```



```
x = 2;
units = 'LY'
switch units
    case {'AU' 'Astronomical  Units'}
        km = 149597871*x
    case  {'LY','lightyear'}
        km = 149597871*63241*x
    case  {'pc' 'parsec'}
        km = 149597871*63241*3.26156*x
    otherwise
        disp('That was an invalid selection.')
end
```

❖ menu 函數建立一個包含按鈕的選單視窗，提供使用者進行選擇，

```
frequency = 123;
units = menu('Select units for output data', 'rad/s','Hz', 'rev/min')
switch units
    case 1
        disp(frequency)
    case 2
        disp(frequency/(2*pi))
    case 3
        disp(frequency*60/(2*pi))
end
```


1.17 MATLAB 中的使用者自訂函數

❖ 第一個函數形式是 m 檔函數，解說如下：

```
function <output_params> = func_name(<input_params>)
<func body>
```

❖ 函數的呼叫型式為：

```
<specific_out_params> = <func_name>(<assigned_input_params>)
```

範例1.1

❖ 鋸齒波的傅利葉級數為

$$y(t) = \frac{1}{2} - \sum_{n=1}^{\infty} \sin\left(\frac{2\pi nt}{T}\right)$$

其中 T 為波形的週期。我們可以建立一個函數來評估給定的 t 和 T 值，雖然無法將無限做加總，可以將 m 項求總和，其中 m 為比較大的值。因此可以定義MATLAB 函數 sawblade 如下，注意此函數具有三個輸入與一個輸出。

範例1.1

```
function y = sawblade(t,T,n_trms)
% Evaluates, at instant t, the Fourier approximation of a sawtooth wave of
% period T using the first n_trms terms in the infinte series.
y = 1/2;
for n = 1:n_trms
    y = y - (1/(n*pi))*sin(2*n*pi*t/T);
end
```

```
c = 1;
for t = 0:0.01:4, y(c) = sawblade(t,2,50); c = c+1; end
plot([0:0.01:4],y)
```

範例1.1

有效的函數呼叫為

```
y = sawblade(0.2*period,period,terms)
```

其中 period 和 term s 為先前指定的值，則

```
y = sawblade(2,5.7,60)
```

或是使用函數 feval

```
y = feval('sawblade',2,5.7,60)
```


範例1.2

❖ 考慮一個進階的範例包含產生矩陣的函數。應用在結構的靜態且/ 或動態分析之有限元素方法的基本特徵，是以矩陣形式表示一小部分或元素的剛性和慣性特性。這些元素矩陣是被用來組合成描述全部的整體結構。知道作用在這些結構的力，我們可以得到結構的靜態或動態反應，一個這樣的元素是個均勻圓軸，對此元素，慣性矩陣與剛性矩陣，角加速度和位移，分別施加的力矩由下式給出

範例1.2

$$\mathbf{K} = \frac{GJ}{L}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

及

$$\mathbf{M} = \frac{\rho JL}{6}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

❖ 其中 L 是軸的長度， G 和 ρ 是材料性質，且 d 是軸的直徑。若我們欲在MATLAB 建立有限元素封裝，其中包含扭轉元素，則需要這些矩陣。下列的函數從軸的特性產生這些矩陣：

範例1.2

```
function [K,M] = tors_el(L,d,rho,G)
J = pi*d^4/32;
K = (G*J/L)*[1 -1;-1 1];
M = (rho*J*L/6)*[2 1;1 2];
```

❖ 注意這函數具有四個輸入參數與兩個輸出矩陣。

❖ 假設我們要定義此函數

$$\left(\frac{x}{2.4}\right)^3 - \frac{2x}{2.4} + \cos\left(\frac{\pi x}{2.4}\right)$$

MATLAB 定義的函數如下：

```
>> f = @(x) (x/2.4).^3-2*x/2.4+cos(pi*x/2.4);
```

```
>> solution = fzero(f,2.9)
solution =
    3.4825
```



```
x = 0:0.1:5; plot(x,f(x))
```

```
>> solution = fzero(@(x) (x/2.4).^3-2*x/2.4+cos(pi*x/2.4), 2.9)
```

❖ 使用者可以輸入 *m* 檔函數或是匿名函數作為一個函數。

```
function y = sp_cubic(x)
y = x.^3-2*x.^2-6;

function [minimum maximum] = minandmax(f,v)
% v is a vector with the start, increment and end value
y = feval(f,v); minimum = min(y); maximum = max(y);
```

❖使用 minandmax 的定義是指 f 可為匿名函數或是 *m* 檔函數，

```
>> [lo hi] = minandmax(f,[-5:0.1:5]);
>> fprintf('lo = %8.4f hi = %8.4f\n',lo,hi)

lo = -181.0000 hi =  69.0000
```

❖使用函數的其他形式為

```
>> [lo hi] = minandmax('sp_cubic',[-5:0.1:5]);
>> fprintf('lo = %8.4f hi = %8.4f\n',lo,hi)

lo = -181.0000 hi =  69.0000
```



```
function [minimum maximum] = minandmax(f,v)
% v is a vector with the start, increment and end value
y = f(v); minimum = min(y); maximum = max(y);
```

❖若 f 為 *m* 檔函數 sp_cubic ，函數 minandmax 就會出現如下的錯誤：

```
>> [lo hi] = minandmax('sp_cubic',[-5:0.1:5]);
>> fprintf('lo = %8.4f hi = %8.4f\n',lo,hi)
??? Subscript indices must either be real positive integers or logicals.

Error in ==> minandmax at 4
y = f(v);
```

1.18 MATLAB 之資料結構

❖ 單元陣列之資料結構是由大括弧{}所定義

```
>> A = cell(4,1);
>> A = {'maths'; 'physics'; 'history'; 'IT'}

A =
    'maths'
    'physics'
    'history'
    'IT'
```



```
>> p = A(2)

p =
    'physics'

>> A(3:4)

ans =
    'history'
    'IT'
```

```
>> cont = A{3}

cont =
history
```

```
>> cont(4)

ans =
t
```

❖ 單元陣列可含數值及字串數據，同時也能由函數 cell 產生。

```
>> F = cell(2,2)

F =
     [ ]     [ ]
     [ ]     [ ]
```


❖ 指定一個純量，陣列或字串至一單元可寫成

```
>> F{1,1} = 2;
>> F{1,2} = 'test';
>> F{2,1} = ones(3);
>> F

F =
    [          2]    'test'
    [3x3 double]       [ ]
```

❖ 另一產生 F 之等效方法是

```
>> F = {[2] 'test'; [ones(3)] [ ]}
```

❖ 使用 celldisp

```
>> celldisp(F)

F{1,1} =
     2

F{2,1} =
     1     1     1
     1     1     1
     1     1     1

F{1,2} =
test

F{2,2} =
     [ ]
```


❖ 設定三個學生之資料在單元陣列：names，fees及subjects 中作為特定值開始。

```
>> names = {'A Best', 'D Good', 'S Green', 'J Jones'}

names =
    'A Best'    'D Good'    'S Green'    'J Jones'

>> fees = {333 450 200 800}

fees =
    [333]    [450]    [200]    [800]
```

```
>> subjects = {'cs','cs','maths','eng'}

subjects =
    'cs'    'cs'    'maths'    'eng'

>> StudentRecords = struct('NameField',names,'FeesField',fees,...
                           'SubjectField',subjects)

StudentRecords =
1x4 struct array with fields:
    NameField
    FeesField
    SubjectField
```



```
>> StudentRecords(1)

ans =
        NameField: 'A Best'
        FeesField: 333
    SubjectField: 'cs'
```

```
>> StudentRecords(1).NameField

ans =
A Best

>> StudentRecords(2).SubjectField

ans =
cs
```

❖ 記錄可以改變或添加最新數據如下：

```
>> StudentRecords(3).FeeField = 1000;
```

```
>> StudentRecords(3).FeeField

ans =
        1000
```


❖ MATLAB 提供一些函數讓我們在數據結構之間轉換，以下列出其中之一部分

```
cell2struct
struct2cell
num2cell
str2num
num2str
int2str
double
single
```

1.19 編寫MATLAB 指令碼

❖ 可以使用 checkcode 來找出錯誤

```
% e3s125.m A script full of errors!!!
A = [1 2 3; 4 5 6
B = [2 3; 7 6 5]
c(1) = 1; c(1) = 2;
for k = 3:9
    c(k) = c(k-1)+c(k-2)
    if k = 3
        displ('k = 3, working well)
end
c
```



```
>> e3s125
Error: File: e3s125.m Line: 3 Column: 3
The expression to the left of the equals sign is not a valid target
                                                    for an assignment.
```

❖ 使用 checkcode 來檢查指令碼 e3s125

```
>> checkcode e3s125
L 3 (C 3): Invalid syntax at '='. Possibly, a ), }, or ] is missing.
L 3 (C 16): Parse error at ']': usage might be invalid MATLAB syntax.
L 5 (C 1-3): Invalid use of a reserved word.
L 7 (C 5-6): IF might not be aligned with its matching END (line 9).
L 7 (C 10): Parse error at '=': usage might be invalid MATLAB syntax.
L 8 (C 15-35): A quoted string is unterminated.
L 11 (C 0): Program might end prematurely (or an earlier error
                                                    confused Code Analyzer).
```

```
% e3s125c.m A script less full of errors!!!
A = [1 2 3; 4 5 6];
B = [2 3; 7 6];
c(1) = 1; c(1) = 2;
for k = 3:9
    c(k) = c(k-1)+c(k-2)
    if k == 3
        disp('k = 3, working well')
    end
end
c
```


❖執行此指令碼

```
>> e3s125c
Attempted to access c(2); index out of bounds because numel(c)=1.

Error in e3s125c (line 6)
    c(k) = c(k-1)+c(k-2)
```

```
>> checkcode e3s125c
L 6 (C 5): The variable 'c' appears to change size on every loop
         iteration (within a script). Consider preallocating for speed.
L 6 (C 10): Terminate statement with semicolon to suppress output
                                                    (within a script).
L 11 (C 1): Terminate statement with semicolon to suppress output
                                                    (within a script).
```

1.20 MATLAB 中一些不明顯的陷阱

- 檔案和函數的命名須小心。
- 不可使用MATLAB 的函數名或指令當作是變數名稱。


```
>> sin = 4

sin =
     4

>> 3*sin

ans =
    12

>> sin(1)

ans =
     4

>> sin(2)
??? Index exceeds matrix dimensions.

>> sin(1.1)
??? Subscript indices must either be real positive integers or logicals.
```

- 矩陣大小是由指派設定。

```
for i = 1:2
    b(i) = i*i;
end
A = [4 5; 6 7];
A*b'
```

- 注意點積。
- 在程式一開始就清除所有變數及設定陣列成空矩陣。

1.21 在MATLAB 裡加快計算速度

範例1.3

❖ 使用 for 迴圈來寫入 **b** 向量的元素

```
% e3s108.m
% Fill b with square roots of 1 to 100000 using a for loop
tic;
for i = 1:100000
    b(i) = sqrt(i);
end
```

```
t = toc;
disp(['Time taken for loop method is ', num2str(t)]);
```

範例1.4

❖使用向量運算來寫入 **b** 向量的元素

```
% e3s109.m
% Fill b with square roots of 1 to 100000 using a vector
tic
a = 1:100000; b = sqrt(a);
t = toc;
disp(['Time taken for vector method is ',num2str(t)]);
```