影像處理

(Image Processing)

真理大學 資訊工程系 吳汶涓老師

Course 2

影像與Matlab

Outline

2.1 灰階影像

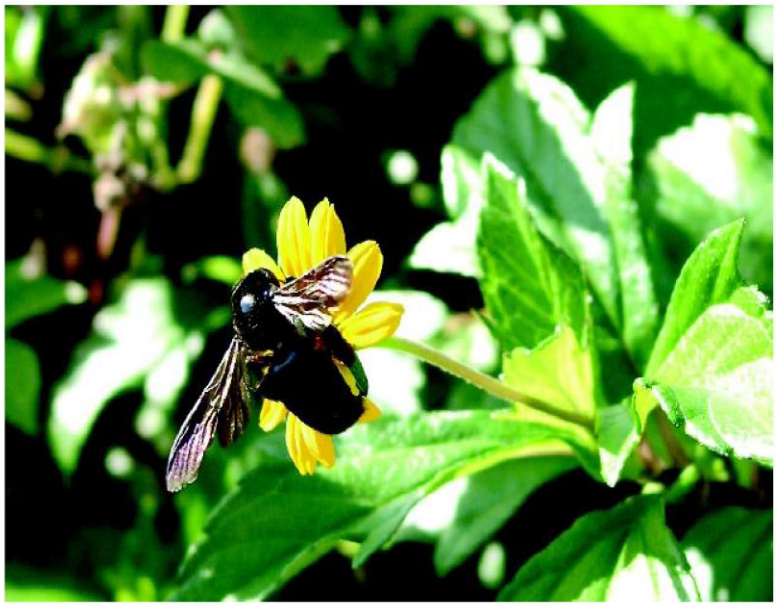
2.2 RGB影像

2.3 索引彩色影像

2.4 資料型態與轉換

2.5 影像檔案與格式

附錄A　Matlab的基本用法



2.1 灰階影像

- 使用Matlab指令視窗開啟灰階影像

```
>> w=imread('wombats.tif');
```

```
>> figure,imshow(w),pixval on
```

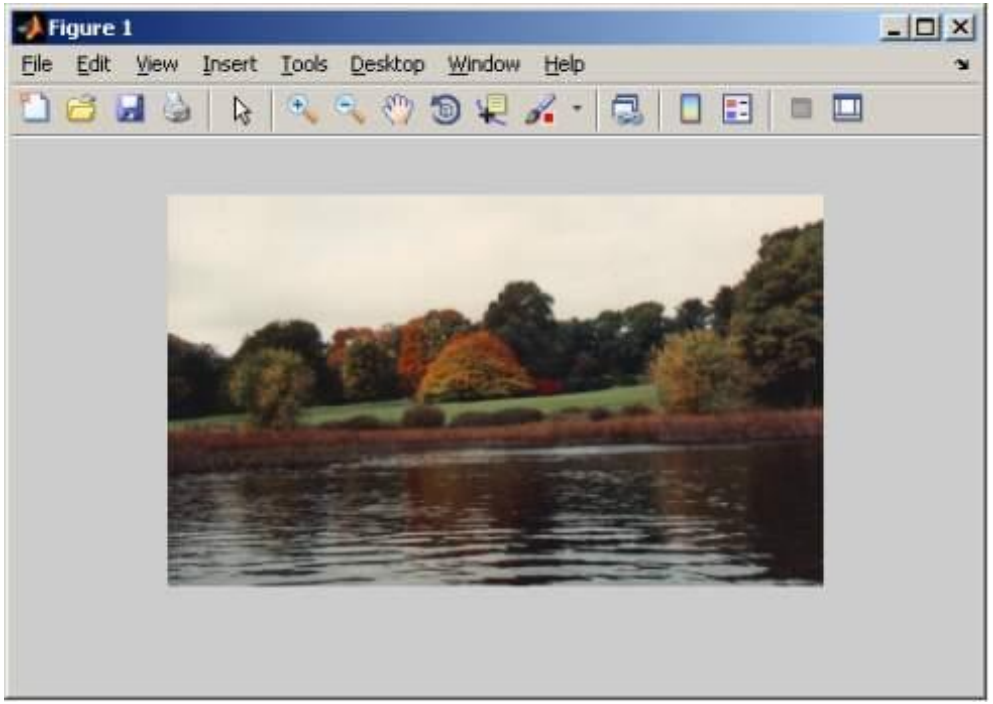
124, 57 = 184
(行) (列) (灰階值)

圖 2.1 執行 pixval on 指令的袋熊影像

2.2 RGB影像

- 使用Matlab指令視窗開啟彩色影像

```
>> a=imread('autumn.tif');
>> figure,imshow(a),pixval on
```



注意：pixval on指令在新的MATLAB版本可能已移除

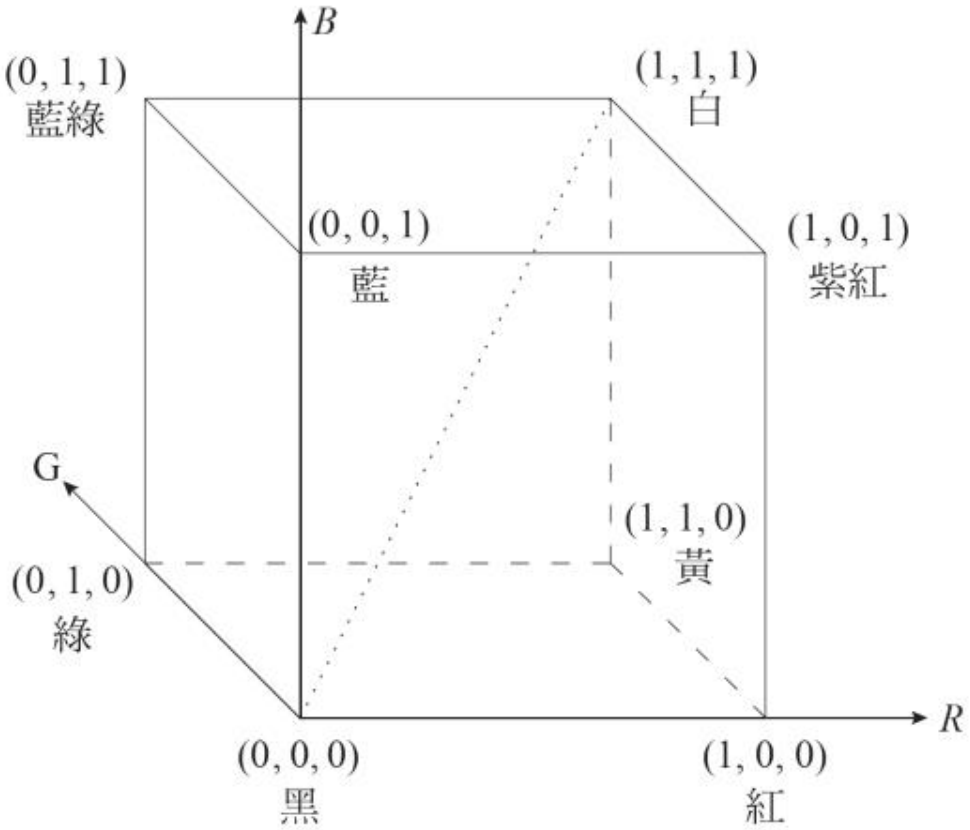


圖 2.2 RGB 色彩模型的色彩立方體

- **多維陣列**（multidimensional array）
  - 透過下面這個指令，可以看出RGB與灰階影像的明顯不同：

```
>> size(a)
```

顯示a的列數、行數、平面數

  - 要獲知任一指定位置的RGB數值，可以使用類似上一節的索引方法。例如：

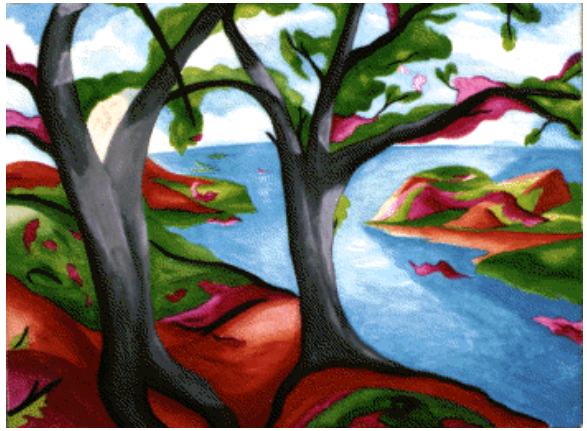
```
>> a(100,200,2)
```

```
>> a(100,200,1:3)
```

2.3 索引彩色影像

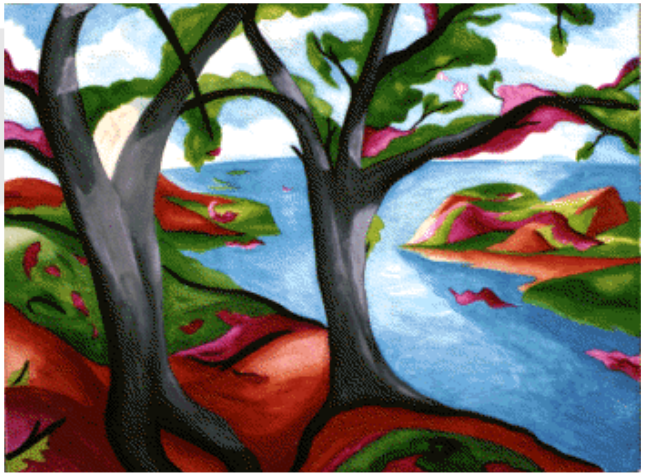
- **索引影像**（indexed image）是包含兩個矩陣：色譜(color map) 與 索引 (index)
  - 像素值並非三個整數

```
>> figure,imshow('emu.tif'),pixval on
```



```
>> em=imread('emu.tif');
>> figure,imshow(em),pixval on
```

```
>> [em,emap]=imread('emu.tif');
>> figure,imshow(em,emap),pixval on
```



- imfinfo函數可以顯示很多影像資訊

```
>> imfinfo('zebra3.bmp')

ans =

                Filename: 'zebra3.bmp'
            FileModDate: '02-Oct-2011 12:52:30'
                FileSize: 45163062
                  Format: 'bmp'
           FormatVersion: 'Version 3 (Microsoft Windows 3.x)'
                   Width: 4752
                  Height: 3168
                BitDepth: 24
               ColorType: 'truecolor'
         FormatSignature: 'BM'
      NumColormapEntries: 0
                Colormap: []
                 RedMask: []
               GreenMask: []
                BlueMask: []
         ImageDataOffset: 54
      BitmapHeaderSize: 40
              NumPlanes: 1
        CompressionType: 'none'
             BitmapSize: 45163008
          HorzResolution: 0
          VertResolution: 0
           NumColorsUsed: 0
       NumImportantColors: 0
```

二元影像的BitDepth=1，但ColorType為 graysacle

2.4 影像與數位影像

- Matlab矩陣中的元素有不同的數字資料型態：

```
>>  a=23;
>> b=uint8(a);
>> b

b =

    23

>> whos a b
  Name        Size              Bytes  Class

  a           1x1                   8  double array
  b           1x1                   1  uint8 array
```

- 型態可以轉換，但只有double能進行數學運算

表 2.1 MATLAB 的資料類型

資料形態	描述	範圍
int8	8 位元整數	−128 ～ 127
uint8	8 位元無號整數	0 ～ 255
int16	16 位元整數	−32768 ～ 32767
uint16	16 位元無號整數	0 ～ 65535
double	雙精度浮點數	與電腦硬體相關

表 2.2 MATLAB 影像轉換

函數	用途	格式
ind2gray	索引轉灰階	y= ind2gray(x,map);
gray2ind	灰階轉索引	[y,map]= gray2ind(x);
rgb2gray	RGB 轉灰階	y=rgb2gray(x);
gray2rgb	灰階轉 RGB	y=gray2rgb(x);
rgb2ind	RGB 轉索引	[y,map]=rgb2ind(x,n);
ind2rgb	索引轉 RGB	y =ind2rgb(x,map);

2.5 影像檔案與格式

- **標頭資訊**（header information）
  - 最少會包含以像素為單位的影像尺寸（長、寬）
  - 還可能包含色譜、壓縮方式及影像的描述
- MATLAB的`imread`和`imwrite`函數目前可支援下列格式：
  - JPEG
  - TIFF
  - GIF
  - BMP
  - PNG
  - PCX
  - ICO
  - CUR
  - HDF
  - XWD

- BMP 圖檔格式
  - 這是微軟公司所提出的點陣圖格式，原本是專門用在Windows作業系統上，讓各軟體的圖形能彼此相容。
  - BMP檔雖然普遍，但有個壞處是無法壓縮全彩圖形，所以存檔後會變的很大（影像佔用多大資料量，存檔後就有多大）。
  - 16色，256色和灰階圖片則可以使用RLE技術壓縮，壓縮後的圖形不會失真，但儲存和開啟的速度會較慢。

- JPEG圖檔格式
  - JPG是網頁常用的圖形格式，他的壓縮率相當驚人，原本1MB的圖片存成JPG檔後可能只剩下幾十K而以（視影像的複雜度及設定的壓縮層級而定）。
  - 由於JPG格式屬於破壞性壓縮，存檔時會捨棄一些不需要的像素（捨棄後就再也就不回來了），因此可能造成圖片失真，不過對一般而言，在正常的壓縮情形下，肉眼是很難看得出壓縮前後的品質差異。
  - 儲存JPG檔時可以選擇壓縮的層級，若選擇高壓縮的方式，則影像的品質會降低; 若選擇高品質的壓縮方式，影像會較接近原來的品質，但檔案也會相對的較大。
  - JPG格式支援RGB全彩，灰階，CMYK模式，但是16色，256色和黑白圖片都無法存成JPG檔。

- GIF圖檔格式
  - GIF是網頁上最常用的圖形格式了，原因是它可以存成透明圖（把其中的一種顏色變透明），交錯圖（在瀏覽器中慢慢顯現），和動畫（把許多GIF圖片連續重疊成一個檔案），而且提供『非破壞性壓縮』，存檔後的體積比原來小很多，圖片也不會失真。
  - GIF最多只能儲存256色的顏色數目，所以在儲存之前，必須將圖片轉為 256色，16色，灰階或黑白模式，才能存成GIF檔。

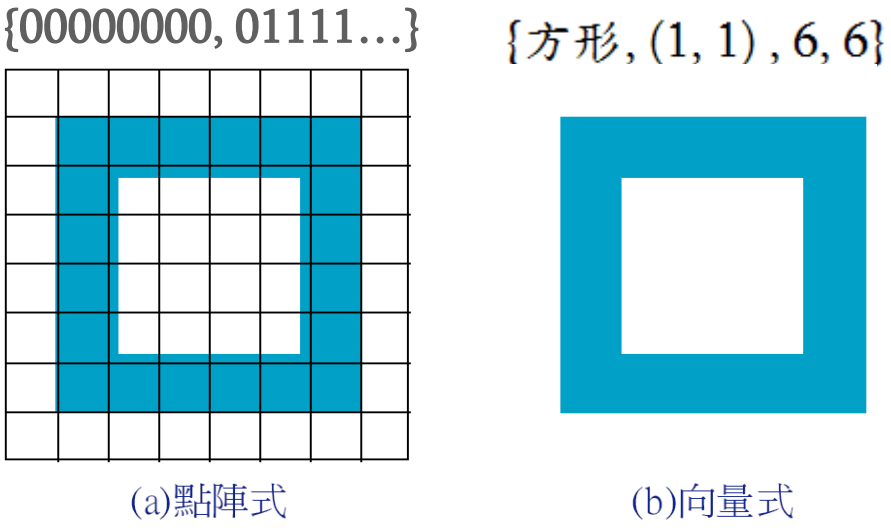
- PNG圖檔格式
  - 近年來由於GIF格式所引發的權利金問題，所以PNG格式就成為免費的GIF替代品，在全球網際網路上作為非破壞性壓縮和顯示影像的格式。
  - 與GIF的差異在於PNG支援全彩影像，可製作透明度的Alpha Channel，但目前尚無法儲存動畫。
  - PNG支援的色彩類型有：RGB全彩，索引色，灰階與黑白模式。
  - PNG格式是W3C Recommendation的網路圖檔格式，目前IE 5.0和Netscape 5.0也都有支援部分PNG格式。
  - PNG格式所包含的規格很多，例如Alpha Channel，Gamma等，若想知道自己的流覽器是否有支援可以連上
    http://www.w3.org/Graphics/PNG

    作個測試就可知道了。

- TIF圖檔格式
  - TIF是影像處理界最普遍之源的檔案格式，因為它可以跨平台，提供非破壞性壓縮，十分適合印刷輸出（因為品質好），所以大多數的影像處理軟體及排版軟體都會支援TIF圖檔。
  - 在一般較為專業的圖庫光碟中，圖檔也都存成TIF格式，以符合排版，印刷的功能。
  - TIF支援RGB全彩，CMYK，索引色彩，灰階和黑白模式。目前也唯有TIF檔能存成48bit的全彩類型，如果有圖片要作為印刷用途，那麼存成TIF檔是最好的方式。

圖檔格式	黑白	16 色	256 色	灰階	全彩	CMYK	壓縮方式	其　　它
BMP	v	v	v	v	v		RLE 非破壞性	只有索引色和灰階 能壓縮
GIF	v	v	v	v			LZW 非破壞性	可存成透明圖、交 錯圖與動畫
JPG				v	v	v	破壞性	可設定壓縮比率
PNG	v	v	v	v	v		非破壞性	新的網頁圖檔格式
PCD			v	v	v			PhotoImpact 無法存 成PCD,但可開啓
TIF	v	v	v	v	v	v	LZW 非破壞性	另支援 16 bit 灰階 與48 bit 全彩
UFO	v	v	v	v	v		非破壞性	可儲存未合併的物 件、路徑及選取區

- 影像儲存方式
  - 向量影像-Object
    - 圖案以物件組成
  - 點陣影像-Pixel
    - 圖案以像素組成



兩種方式都可用來表示直線影像

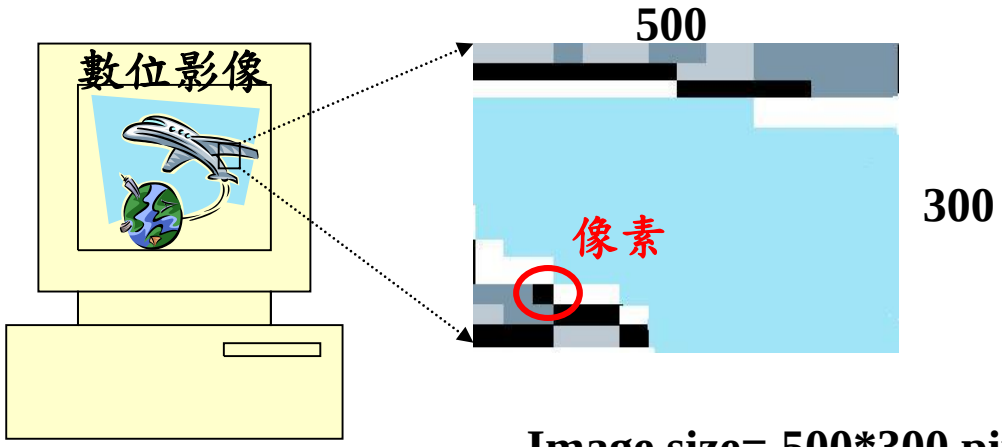


Image size= 500*300 pixels = ? bits

BMP影像的檔案標頭

```
>> dumphex('blocksets.bmp',4)

ans =

42 4D 6E 18 00 00 00 00 00 00 36 00 00 00 28 00  BMn.......6...(.
00 00 42 00 00 00 1F 00 00 00 01 00 18 00 00 00  ..B.............
00 00 38 18 00 00 C4 0E 00 00 C4 0E 00 00 00 00  ..8.............
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

- 這個影像的寬度資料位於第18 至21 位元組，在第二列：
  42 00 00 00
  要知道確實的寬度，必須將這些位元組重新排列：
  00 00 00 42
  轉乘十進位後，寬度為$(4 \times 16^1)+ (2 \times 16^0) = 66$

- 影像高度 1F 00 00 00
  $(1 \times 16^1) + (F \times 16^0) = 16 + 15 = 31$

位元組	資訊	描述
0 ～ 1	格式簽章	ASCII 字元 BM，或十六進位制數字 42 4D。
2 ～ 5	檔案大小	以位元組為單位的檔案大小。
6 ～ 9	保留	全零。
10 ～ 13	資料偏移值	影像掃描點資料儲存起點相對於檔頭的位置。
14 ～ 17	大小	資訊標頭的大小＝ 40 位元組。
18 ～ 21	寬	影像的寬度（像素）。
22 ～ 25	高	影像的高度（像素）。
26 ～ 27	平面	影像平面數目（＝ 1）。
28 ～ 29	位元計數	每個像素的位元數： 1：二元影像，兩種顏色， 4：2^4 ＝ 16 色（索引）， 8：2^8 ＝ 256 色（索引）， 16：16 位元 RGB；2^{16} ＝ 65,536 色， 24：24 位元 RGB；2^{24} ＝ 17,222,216 色。

位元組	資訊	描述
30 ～ 33	壓縮方式	使用的壓縮方法： 0：未壓縮（最常使用）， 1：8 位元 RLE 編碼（極少使用）， 2：4 位元 RLE 編碼（極少使用），
34 ～ 37	影像大小	影像的大小，若壓縮值為 0，則此數值可能為 0。
38 ～ 41	水平解析度	水平方向每一公尺的像素解析度。
42 ～ 45	垂直解析度	垂直方向每一公尺的像素解析度。
46 ～ 49	使用色彩	影像中使用的顏色數目。若此數值為 0，則色彩數目就是每個像素所占位元能容納的最多色彩，也就是 2 位元計數。
50 ～ 53	重要色彩	影像中重要色彩的數目。若所有顏色都很重要，則此數值會設為 0。

- 使用imwrite函數將影像矩陣寫入影像檔案
  - 格式：儲存影像矩陣X及色譜map（如果有的話）寫入使用格式fmt的檔案名稱filename

```
imwrite(X,map,'filename','fmt')
```

  - 若沒有map引數，則影像資料就會被判斷為灰階或RGB

```
a=imread('autumn.tif');
imwrite(a,'autumn.png','png');
```

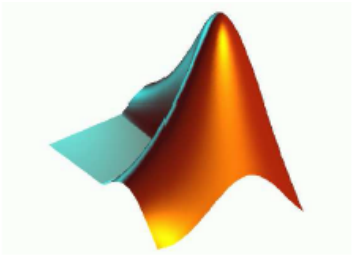
練習

- 將一個灰階影像，儲存成JPEG、PNG及BMP的檔案格式
- 將一個灰階影像，儲存成二元影像、索引色彩影像及全彩影像

附錄A Matlab的基本用法

- 參考用書：
  『MATLAB程式設計入門篇』(第三版) / 張智星 著，碁峰資訊，2011.
  http://mirlab.org/jang/

MATLAB 是什麼？



- MATLAB為美國Mathworks公司於1984年所推出的數學運算軟體。
- 其名稱是由「矩陣實驗室」(MATrix LABoratory)所合成。
- 特長於矩陣相關運算及各領域數值問題。
- 提供科學計算、數值分析、圖形繪製、系統模擬、訊號處理等功能
- 為各種動態系統模擬、數位訊號處理、科學計算、科學目視等領域的標準程式語言。

MATLAB 小傳

- MATLAB早在 1978 年即已現身，是用 Fortran 撰寫的免費軟體，其作者是當時任教於新墨西哥大學的 Cleve Moler 教授。
- Jack Little（又稱為 John Little）將 MATLAB 以C語言重寫，並於 1984 年成立 MathWorks公司，首次推出 MATLAB 商用版。(DOS 版)
- 1993年進入Windows作業系統，推出Simulink互動式動態模擬環境以及符號數學運算功能。
- 之後不斷改版、改版、再改版，目前最新版本為R2015a，其網站為http://www.mathworks.com

為什麼要用 MATLAB？

- MATLAB是個直譯式高階語言，和其他常見的C/C++、JAVA及VB等高階語言比較起來，MATLAB在程式撰寫及資訊視覺化視窗這兩方面相當方便，初學者可說是一學就會、入門輕鬆。
- MATLAB省略許多複雜的語法，採取接近人類思維的語法，同時提供許多指令處理複雜運算。
- MATLAB目前已被廣泛應用於數學、工程、物理、化學、醫學、金融、生物資訊等領域有關數值計算問題。
- 許多國內外大學教科書將納為問題求解模擬示範的工具軟體。

使用 MATLAB 的好處

- MATLAB是一種可以提供便利的環境用以進行各種數學運算的電腦程式
  - 直譯式的數學程式語言，不需編譯(compile)或連結(link)即可直接執行
  - 將所有變數均存成double的形式，所以不需經過變數宣告，亦不需要陣列宣告
  - 包含了非常豐富的內建函數，同時亦提供完整的線上輔助說明

附錄A Matlab的基本用法

- MATLAB 是一套支援矩陣與矩陣運算的資料分析套裝軟體。
  - **指令視窗** (command window)

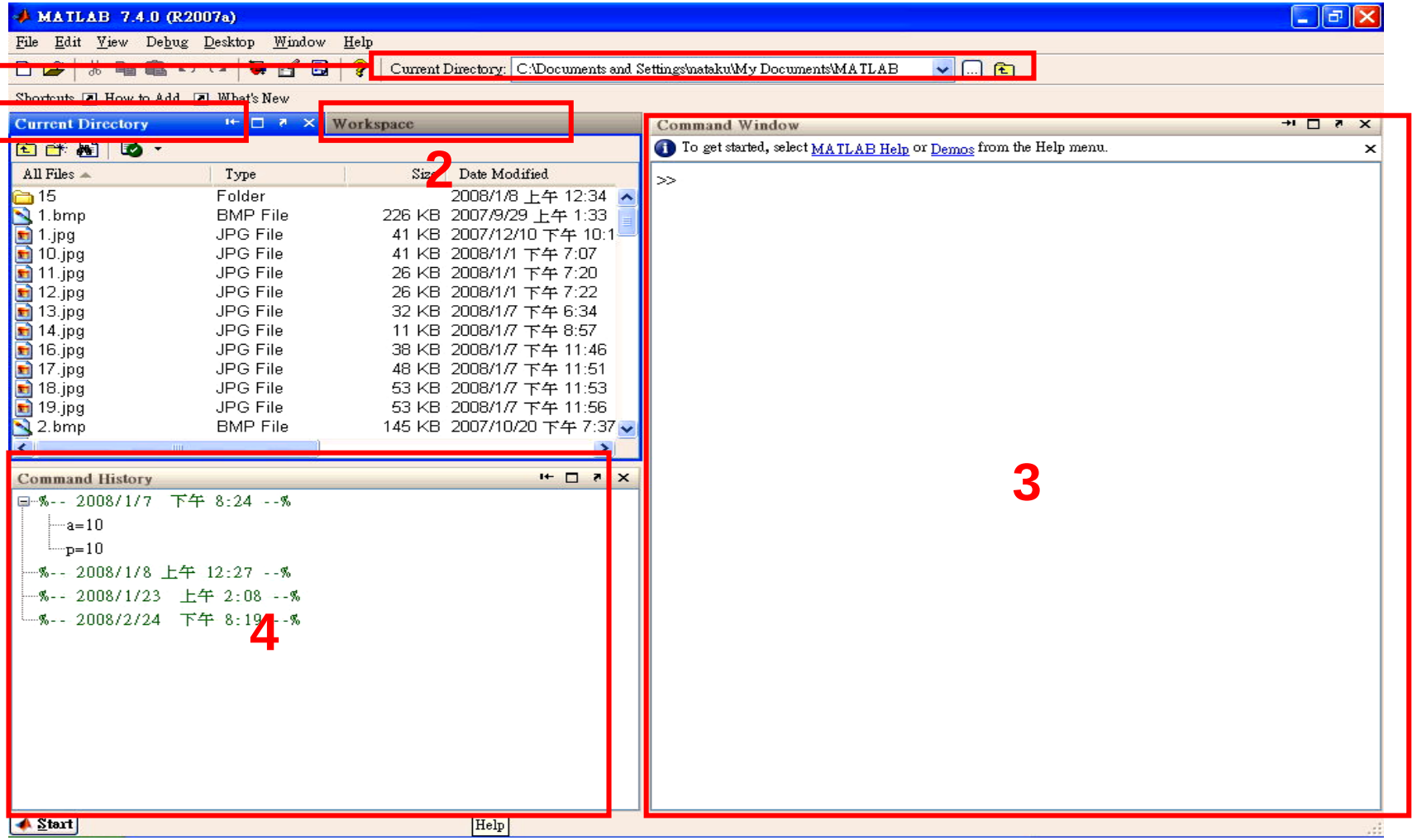
```
>>
```

  - 鍵入指令：

```
>> w=imread('wombats.tif');
```

```
>> figure, imshow(w), pixval on
```

介面說明


MATLAB 7.4.0 (R2007a)
File Edit View Debug Desktop Window Help
Current Directory: C:\Documents and Settings\nataku\My Documents\MATLAB
Shortcuts How to Add What's New
Current Directory
Workspace
All Files Type Size Date Modified
15 Folder 2008/1/8 上午 12:34
1.bmp BMP File 226 KB 2007/9/29 上午 1:33
1.jpg JPG File 41 KB 2007/12/10 下午 10:1
10.jpg JPG File 41 KB 2008/1/1 下午 7:07
11.jpg JPG File 26 KB 2008/1/1 下午 7:20
12.jpg JPG File 26 KB 2008/1/1 下午 7:22
13.jpg JPG File 32 KB 2008/1/7 下午 6:34
14.jpg JPG File 11 KB 2008/1/7 下午 8:57
16.jpg JPG File 38 KB 2008/1/7 下午 11:46
17.jpg JPG File 48 KB 2008/1/7 下午 11:51
18.jpg JPG File 53 KB 2008/1/7 下午 11:53
19.jpg JPG File 53 KB 2008/1/7 下午 11:56
2.bmp BMP File 145 KB 2007/10/20 下午 7:37
Command History
%-- 2008/1/7 下午 8:24 --%
a=10
p=10
%-- 2008/1/8 上午 12:27 --%
%-- 2008/1/23 上午 2:08 --%
%-- 2008/2/24 下午 8:19 --%
Command Window
To get started, select MATLAB Help or Demos from the Help menu.
>>
Start
Help
1
2
3
4

1. Current Directory

—目前檔案所存放的路徑

2. Workspace

—顯示目前工作空間中，所有的變數與其數值

3. Command Window

—輸入指令與資料之介面

4. Command History

—顯示層經在Command Window所用過的指令

- 在命令列中，打上

```
>> 2+2
>> 3*4
>> 2^5
>> sqrt(6)
>> sin(pi/8)
>> log(10)
>> log10(2)
>> a= sin(pi/9)-cos(pi/9)
```

變數名稱限制

- 變數名稱小於等於19字元
- 第一字元不能為數字
- 字元大小寫表示不同意義
- **Ex1: c123（o）**
  4c123（X）
  c_123（o）
  c-123（X）

註：MATLAB 在使用變數時，不需預先經過變數宣告（Variable Declaration）的程序，而且所有數值變數均以預設的 double 資料型式儲存。

- **分號(；)：**

  1.置放於指令結尾處，使其不顯示執行過 程與結果，但仍會執行輸入之指令

  2.在輸入陣列資料時，用於每一個列(row)的結尾

  範例：A=[1,2,3;4,5,6]

$$A=\begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\end{bmatrix}$$

- **加 減 乘 除(＋、－、×、÷)：基本四則運算符號**
- Matlab的標準數據資料型態為 double
- who and whos: 顯示目前工作空間定義的變數

Getting workspace information

- who　　%顯示變數名稱
- whos　　%顯示變數名稱及其大小格式
- dir　　%顯示所有檔案
- what　　%顯示檔案＊.m或＊.mat
- clc　　%清除螢幕
- clear　　%清除變數

>>clear all　　%清除所有變數

>>clear a b　　%清除變數 a and b

常見數學函數

1. *abs(x)* % 取絕對值
2. *acos(x)* % $\cos^{-1}(x)$
3. *acosh(x)* % $\cosh^{-1}(x)$
4. *angle(x)* % 複數的角度
5. *asin(x)* % $\sin^{-1}(x)$
6. *atan(x)* % $\tan^{-1}(x)$
7. *atanh(x)* % $\tanh^{-1}(x)$
8. *ceil(x)* % 取最接近且大於原數的整數（無條件進入）
9. *floor(x)* % 取最接近且小於原數的整數
10. *round(x)* % 四捨五入（取至整數為止）
11. *fix(x)* % 無條件捨去

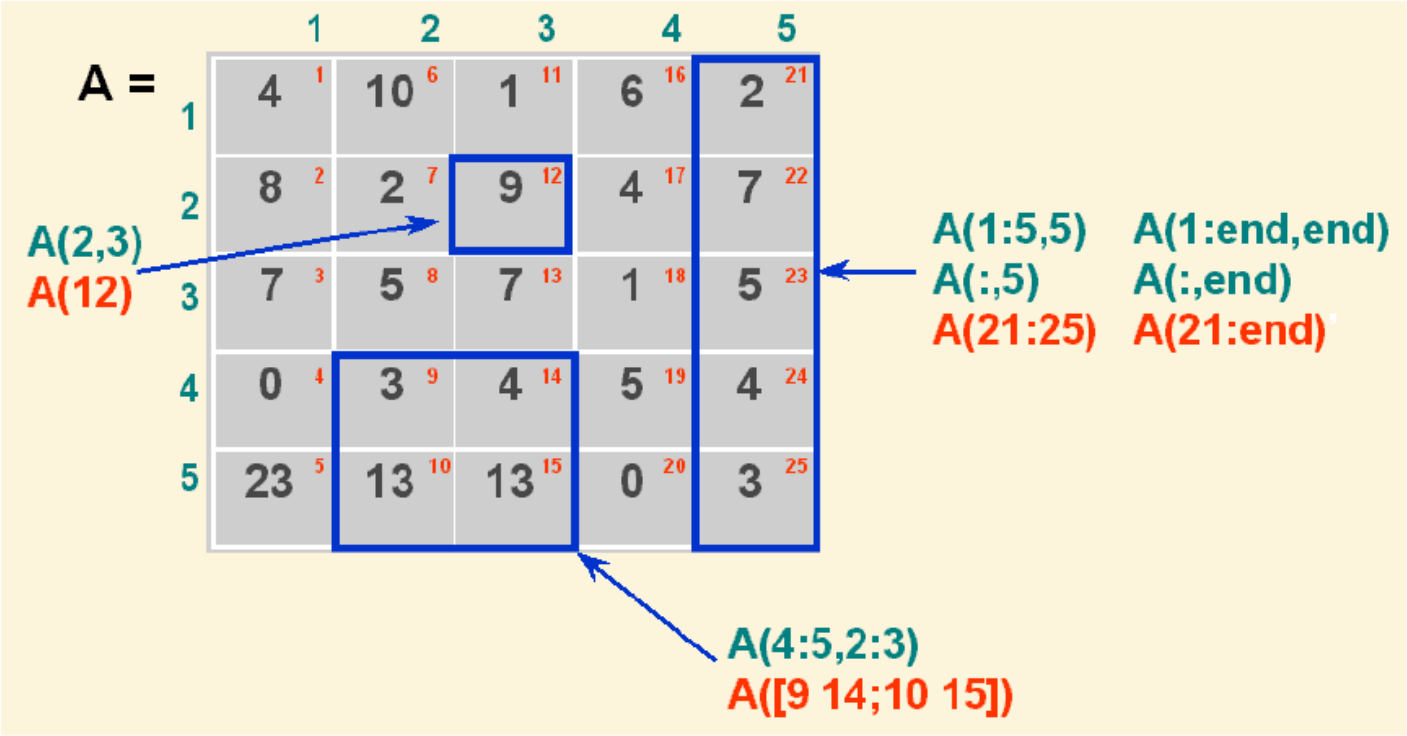
常見數學函數

12. *conj(x)* % 共軛複數
13. *cosh(x)* % cosine hyperbolic function
14. *exp(x)* % exponential : e^x
15. *real(x)* % 取實部 *imag(x)* 取虛部
16. *log(x)* % $\log_e x = \ln x$
17. *log10(x)* % $\log_{10} x$
18. *rem(x,y)* % x/y的餘數
19. *sign(x)* % 取正負號
20. *sin(x)*
21. *sqrt(x)* % $\sqrt{x}$
22. *tan(x)*

- 處理矩陣
  - 變數基本上是矩陣的結構
    a=[2 -2 1; 3 3 9; 4 2 8]
  - 取得矩陣類元素
    >>a(2,1)
    ans = 3
    >>a(6)
    ans = 2
    >>a([3 4 7])
    ans = 4 -2 1

$$\begin{bmatrix} 2 & -2 & 1 \\ 3 & 3 & 9 \\ 4 & 2 & 8 \end{bmatrix}$$

矩陣的索引或下標


A =
1 2 3 4 5
1 4 1 10 6 1 11 6 16 2 21
2 8 2 2 7 9 12 4 17 7 22
3 7 3 5 8 7 13 1 18 5 23
4 0 4 3 9 4 14 5 19 4 24
5 23 5 13 10 13 15 0 20 3 25
A(2,3)
A(12)
A(1:5,5)
A(:,5)
A(21:25)
A(1:end,end)
A(:,end)
A(21:end)
A(4:5,2:3)
A([9 14;10 15])

- 冒號(：)
  - 產生數列

    格式：起始：公差：結束 或 起始:結束

    範例　A=1:2:10

    A=[1 3 5 7 9 10]
  - 應用於矩陣

    >> a(2, 1:2)

    ans = 3 3

    >> a(1:3, 1)

    ans = 2 3 4

$$\begin{bmatrix} 2 & -2 & 1 \\ 3 & 3 & 9 \\ 4 & 2 & 8 \end{bmatrix}$$

- 矩陣運算

```
>>a=[4 -2 -4 7; 1 5 -3 2; 6 -8 -5 -6; -7 3 0 1];
>>b=[2 4 -7 -4; 5 6 3 -2; 1 -8 -5 -3; 0 -6 7 -1];
>>c=2*a-3*b
c = 2  -16   13   26
   -13  -8  -15  10
    9    8    5   -3
   -14  24  -21   5
```

  - 反轉、轉置矩陣

```
>>inv(a)
>>a'
```

矩陣運算相關指令

```
flipud(a)          %上下顛倒
fliplr(a)          %左右顛倒
rot90(a)           %旋轉90度(逆時針)
reshape(a,m,n)     %重定矩陣行列數
diag(v)            %取對角線元素所形成之向量
inv(A)             %反矩陣
eig(A)             %特徵值
rank(A)            %秩、階數
[r,c]=size(A)
n=length(v)
```

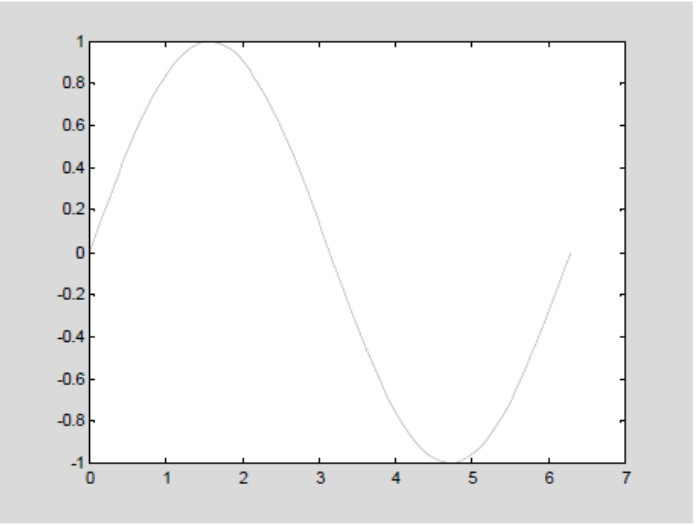
一些特殊矩陣

1. *eye(n,m)* %單位矩陣nxm
2. *eye(n)* %單位矩陣nxn
3. *ones(n,m)* %常數矩陣nxm全部爲1
4. *ones(n)* %常數矩陣nxn全部爲1
5. *zeros(n,m)* %常數矩陣nxm全部爲0
6. *zeros(n)* %常數矩陣nxn全部爲0
7. *rand(n,m)* %亂數所形成nxm的矩陣
8. *randn(n)* %亂數所形成nxn的矩陣

基本的繪圖指令

- plot：最基本的繪圖指令
- 對 x 座標及相對應的 y 座標進行作圖

```
x = linspace(0, 2*pi);      % 在 0 到 2π 間，等分取 100 個點
y = sin(x);                 % 計算 x 的正弦函數值
plot(x, y);                 % 進行二維平面描點作圖
```

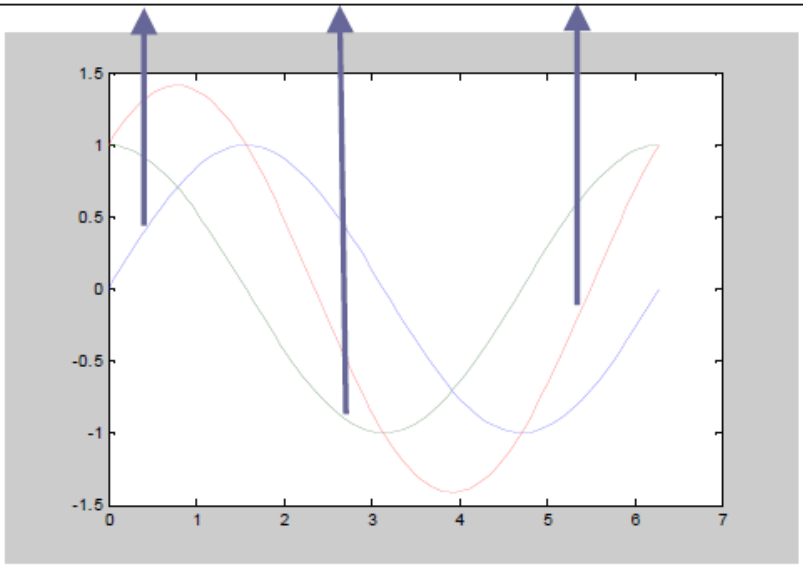


註：只給定一個向量，則以該向量則對其索引值(Index)作圖

繪多條曲線於同一圖

- 方法一：將 x 及 y 座標依次送入plot 指令

```
x = linspace(0, 2*pi);                % 在 0 到 2π 間，等分取 100 個點
plot(x, sin(x), x, cos(x), x, sin(x)+cos(x));  % 進行多條曲線描點作圖
```



- 畫出多條曲線時，會自動輪換曲線顏色

- 方法二：疊圖

```
plot(x, sin(x)); hold on; plot(x, cos(x)); hold off;
```

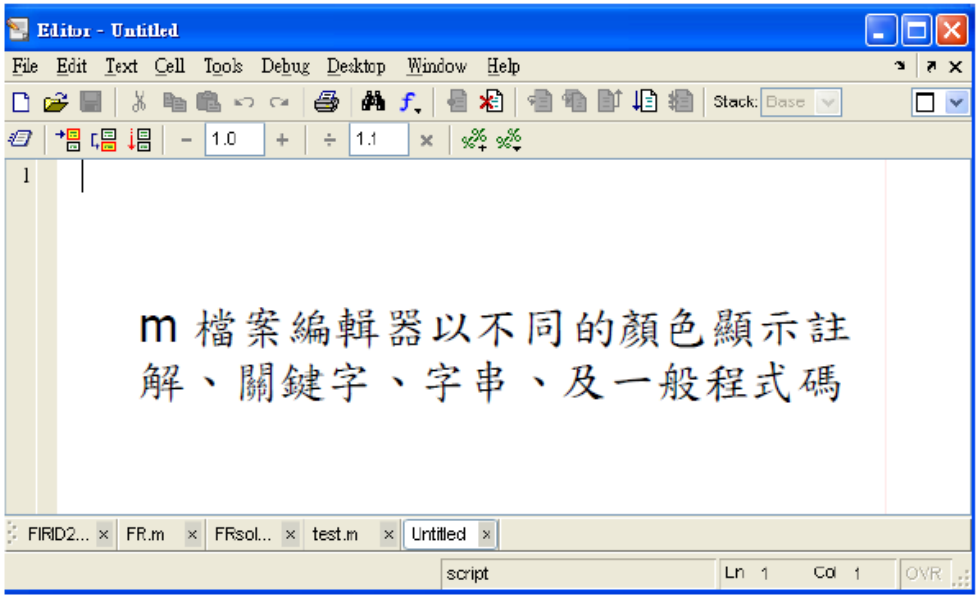
MATLAB 之程式類型 (m檔案)

- 一個用來執行MATLAB的最普遍方式，就是在命令視窗中一次輸入一個指令。m 檔提供另一條進行運算的途徑，可以擴展MATLAB解決問題的能力。一個**m檔** (m-file) 包括一系列可以在同一時間執行的敘述。

- M檔有兩種形式，副檔名皆為m (filename.m)
  - 腳本檔案 (script file)：被用來保留一連串的命令
  - 函數檔案 (function file)

m 檔案編輯器

- m 檔案是文字檔
  - 可以用各種文字編輯器修改
- MATLAB 提供了內建的「m 檔案編輯器」（m-file editor）
  - 點選指令視窗的 file/open 下拉式選單，開啟 m 檔案編輯器
  - 或在指令視窗直接鍵入「edit filename.m」或「open filename.m」



函數檔案

- 所謂**函數檔案** (function file) 就是以**function**這個文字起頭的m檔。
- 函數檔案可以接受引數並且傳回輸出值。即可接受**輸入變數**，並將結果送至**輸出變數**。

```
function outvar = funcname(arglist)
  % help comments
  statements
  outvar = value;
```

```
function average = func1(vector)
average = sum(vector)/length(vector);        % 計算平均值
```

函數 (Function)

- 用 type 指令顯示其內容：
  - >> **type func1.m**

```
function average = func1(vector)
average = sum(vector)/length(vector);        % 計算平均值
```

- 第一列為函數定義列（Function Definition Line）
  - 定義函數名稱（func1，最好和檔案的檔名相同）
  - 輸入引數（vector）
  - 輸出引數（average）
  - function為關鍵字
- 第二列以後為函數主體（Function Body）
  - 規範函數運算過程，並指定輸出引數的值

函數的呼叫

- 呼叫的基本語法 (一個函數可以有多輸入及輸出)

  [Output1, Output2, …] = funcname(input1, input2, …)

  - func1之呼叫方式

```
>> vec = [1 5 3];
>> ave = func1(vec)
ave =
    3
```

  - func2.m 可接受兩個輸入並產生兩個輸出

```
function [ave1, ave2] = func2(vector1, vector2);
ave1 = sum(vector1)/length(vector1);
ave2 = sum(vector2)/length(vector2);
```

  - func2.m 的呼叫方式

```
>> [a, b] = func2([1 2 3], [4 5 6 7 8])
a =
    2
b =
    6
```

迴圈指令

- MATLAB 提供兩種迴圈指令，一種是 for 迴圈，另一種是 while 迴圈
  - **for迴圈** (for loop) 在進行**指定次數**的重複動作之後停止。
  - **while迴圈** (while loop) 則在**某一個邏輯條件成立**時終止。

```
for  變數 = 向量
     運算式
end
```

```
for  變數1 = 向量1
     運算式1
     for  變數2 = 向量2
          運算式2
     end
      :
      :
end
```

```
while  條件式
     運算式
end
```

for 迴圈範例

```
for  i=1:3
       y(i)=cos(i)
end
```

執行結果

```
y =
   0.5403
y =
   0.5403   -0.4161
y =
   0.5403   -0.4161   -0.9900
```

Ex: 1+2+3+4+5...+10=?

```
sum=0;
for  i=1:10;
   sum=sum+i;
end
fprintf('sum= %5.0f \n', sum)
```

執行結果

```
sum =    55
```

- 若要跳出 for 迴圈，可用 **break** 指令

while 迴圈範例

Ex: 1+2+3...+n>50 最小之n值?

程式

```
sum=0;
n=0;
while sum<=50
    n=n+1;
    sum=sum+n;
end
fprintf('1+2+...+n >50 最小之n值= %3.0f \n', n)
```

執行結果

```
1+2+...+n >50 最小之n值=  10
```

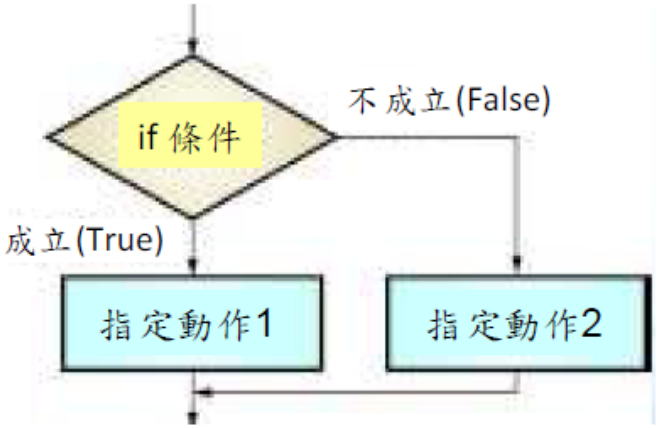
- 若要跳出 while 迴圈，亦可用 **break** 指令

條件指令

- MATLAB 支援二種條件指令，一種是 if-else-end 條件指令，另一種是 MATLAB 在第五版之後開始支援的 switch - case - otherwise 條件指令

- 最常用的條件指令是 if - else - end，其使用語法為：

```
if  條件式
    運算式一
else
    運算式二
end
```



if - else - end 範例

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

```
y(1) = 0 is even.
y(2) = 3 is odd.
y(3) = 4 is even.
y(4) = 1 is odd.
y(5) = 6 is even.
```

- 上述的 if - then - else 為雙向條件，亦即程式只會執行「運算式一」或「運算式二」，不會有第三種可能

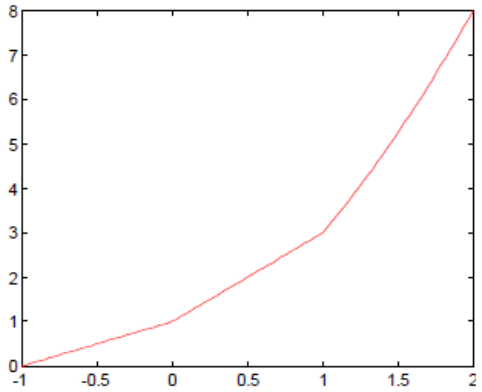
多向條件指令

- MATLAB 亦可執行多向條件，若要進行更多向的條件，只需一再重覆 **elseif** 即可

```
if  條件式一
    運算式一
elseif  條件式二
    運算式二
else
    運算式三
end
```

Ex. $f(x)=\begin{cases} x+1 & ,x\le 0 \\ 2x+1 & ,0<x\le 1 \\ x^2+2x & ,1<x\le 2 \end{cases}$ plot f(x) v.s. x

```
x=linspace(-1,2,100);
for  i=1:length(x)
   if  x(i)<=0
     y(i)=x(i)+1;
   elseif  x(i)<=1
     y(i)=2*x(i)+1;
   else
     y(i)=x(i)^2+2*x(i);
   end
end
plot(x,y)
```



switch-case-otherwise 範例

- 欲根據月份來判斷其季別，可輸入如下：

```
for month = 1:12
        switch month
                case {3,4,5}
                        season = 'Spring';
                case {6,7,8}
                        season = 'Summer';
                case {9,10,11}
                        season = 'Autumn';
                case {12,1,2}
                        season = 'Winter';
        end
        fprintf('Month %g ===> %s.\n', month, season);
end
```

```
switch   expression
 case   value 1
    statement 1
 case   value 2
    statement 2
 case   value n-1
    statement n-1
 otherwise
    statement n
 end
```

```
Month 1 ===> Winter.
  :                :
  :                :
  :                :
Month 12 ===> Winter.
```

Exercise

- x=[1:10]，f(x)=sin(x)，請分別利用 for 迴圈與 while 迴圈計算這些 x 點的函數值，以一個向量表示之。
- 若 x=[0:0.5:5]，請重做上題。
- 完成一個 5x4 的矩陣，其中各元素之值為該元素行索引值加上列索引值的和。
- 寫一個 MATLAB 的遞迴函數 fibo.m 來計算 Fibonacci 數列，其定義如下：

  fibo(n+2) = fibo(n+1)+fibo(n)

  此數列的啟始條件如下：

  fibo(1) = 0, fibo(2) = 1