

MATLAB 程式設計入門篇

初探MATLAB

張智星 清大資工系

補充內容：方煒 台大生機系

2-1 使用變數與基本運算

■ 一般數學符號運算

- 在MATLAB 命令視窗（Command Window）內的提示符號（>>）之後輸入運算式，並按入 Enter 鍵即可。例如：

```
>> (5*2+3.5)/5  
ans =  
    2.7000
```

```
>> 1+1  
>> 2*3  
>> 5.23e12/6.2e5  
>> exp(-3)  
>> atan2(-1,2)  
>> sin(5) ← In radians
```

- 若不想讓 MATLAB 每次都顯示運算結果，只需在運算式最後加上分號（;）即可，例如：

```
>> (5*2+3.5)/5;
```

變數命名規則與使用

- 第一個字母必需是英文字母。
- 字母間不可留空格。
- 最多只能有 31 個字母，MATLAB 會忽略多餘字母（在 MATLAB 第 4 版，則是 19 個字母）。
- 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。

加入註解

- 若要加入註解（Comments），可以使用百分比符號（%）例如：

```
>> y = (5*2+3.5)/5; % 將運算結果儲存在變數 y，但不用顯示於螢幕  
>> z = y^2          % 將運算結果儲存在變數 z，並顯示於螢幕  
z =  
    7.2900
```

2-2 向量與矩陣的處理

- MATLAB 中的變數還可用來儲存向量（Vectors）及矩陣（Matrix），以進行各種運算，例如：

```
>> s = [1 3 5 2];% 注意 [] 的使用，及各數字間的空白間隔
```

```
>> t = 2*s+1
```

```
t =
```

```
3    7   11    5
```

矩陣的各種處理

- MATLAB 亦可取出向量中的一個元素或一部份來做運算，例如：

```
>> t(3) = 2 % 將向量 t 的第三個元素更改為 2
```

```
t =
```

```
3 7 2 5
```

```
>> t(6) = 10 % 在向量 t 加入第六個元素，其值為 10
```

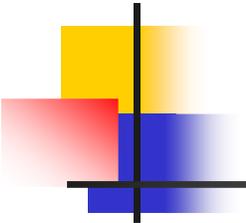
```
t =
```

```
3 7 2 5 0 10
```

```
>> t(4) = [] % 將向量 t 的第四個元素刪除，[] 代表空集合
```

```
t =
```

```
3 7 2 0 10
```



建立大小為 $m \times n$ 的矩陣

- 在每一橫列結尾加上分號 (;) ，例如：

```
>> A = [1 2 3 4; 5 6 7 8; 9 10 11 12]; % 建立 3x4 的矩陣 A
```

```
>> A % 顯示矩陣 A 的內容
```

```
A =
```

```
1    2    3    4
5    6    7    8
9   10   11   12
```

$m \times n$ 矩陣的各種處理之一

- `>> A(2,3) = 5` % 將矩陣 A 第二列、第三行的元素值，改變為 5

A =

1	2	3	4
5	6	5	8
9	10	11	12

- `>> B = A(2,1:3)` % 取出矩陣 A 的第二橫列、第一至第三直行，並儲存成矩陣 B

B =

5	6	5
---	---	---

$m \times n$ 矩陣的各種處理之二

- `>> A = [A B']` % 將矩陣 B 轉置後、再以行向量併入矩陣 A

A =

```
1  2  3  4  5
5  6  5  8  6
9 10 11 12  5
```

- `>> A(:, 2) = []` % 刪除矩陣 A 第二行（：代表所有橫列，[]代表空矩陣）

A =

```
1  3  4  5
5  5  8  6
9 11 12  5
```

$m \times n$ 矩陣的各種處理之三

- `>> A = [A; 4 3 2 1]` % 在原矩陣 A 中，加入第四列

A =

1	3	4	5
5	5	8	6
9	11	12	5
4	3	2	1

- `>> A([1 4], :) = []` % 刪除第一、四列（：代表所有直行，[]是空矩陣）

A =

5	5	8	6
9	11	12	5

2-3 常用數學函數

- MATLAB 是一個科學計算軟體，因此可以支援很多常用到的數學函數
 - `>> y = abs(x) % 取 x 的絕對值`
 - `>> y = sin(x) % 取 x 的正弦值`
 - `>> y = cos(x) % 取 x 的餘弦值`
 - `>> y = tan(x) % 取 x 的正切值`
 - `>> y = exp(x) % 自然指數 exp(x)`
 - `>> y = log(x) % 自然對數 ln(x)`

```
sin([pi/4,pi/2,pi])= [0.7071 1.0000 0.0000]
```

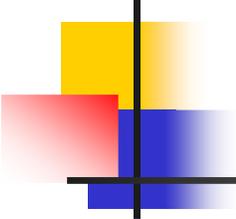
數學函數

- $\cos(x)$ $\sin(x)$ $\tan(x)$ $\sec(x)$ $\csc(x)$ $\cot(x)$
- $\arccos(x)$ $\arcsin(x)$ $\arctan(x)$ $\operatorname{atan2}(y,x)$
- $\exp(x)$ $\log(x)$ [$\log(x)$ is $\ln(x)$] $\log_{10}(x)$ $\log_2(x)$ \sqrt{x}
- $\cosh(x)$ $\sinh(x)$ $\tanh(x)$ $\operatorname{sech}(x)$ $\operatorname{csch}(x)$ $\operatorname{coth}(x)$
- $\operatorname{acosh}(x)$ $\operatorname{asinh}(x)$ $\operatorname{atanh}(x)$
- $\operatorname{sign}(x)$ $\operatorname{airy}(n,x)$
- $\operatorname{besselh}(n,x)$ $\operatorname{besseli}(n,x)$ $\operatorname{besselj}(n,x)$ $\operatorname{besselk}(n,x)$
- $\operatorname{bessely}(n,x)$ $\operatorname{beta}(x,y)$ $\operatorname{betainc}(x,y,z)$ $\operatorname{betaln}(x,y)$
- $\operatorname{ellipj}(x,m)$ $\operatorname{ellipke}(x)$
- $\operatorname{erf}(x)$ $\operatorname{erfc}(x)$ $\operatorname{erfcx}(x)$ $\operatorname{erfinv}(x)$ $\operatorname{gamma}(x)$ $\operatorname{gammainc}(x,a)$
- $\operatorname{gammainl}(x)$ $\operatorname{expint}(x)$ $\operatorname{legendre}(n,x)$ $\operatorname{factorial}(x)$

常用函數

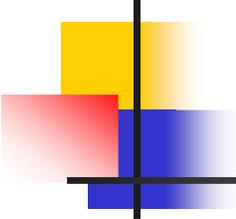
abs(x)	the absolute value of a number (real or complex)
clc	clears the command window
ceil(x)	the nearest integer to x looking toward +1 clear clears all assigned variables
close all	closes all figure windows
close 3	closes figure window 3
fix(x)	the nearest integer to x looking toward zero
fliplr(A)	flip a matrix A, left for right
flipud(A)	flip a matrix A, up for down
floor(x)	the nearest integer to x looking toward -1
length(a)	the number of elements in a vector
mod(x,y)	the integer remainder of x/y; see online help if x or y are negative
rem(x,y)	the integer remainder of x/y; see online help if x or y are negative
rot90(A)	rotate a matrix A by 90
round(x)	the nearest integer to x
sign(x)	the sign of x and returns 0 if x=0
size(c)	the dimensions of a matrix

floor([1.5,2.7,-1.5])



fprintf

- `fprintf(' N =%g \n',500)`
- `fprintf(' x =%1.12g \n',pi)`
- `fprintf(' x =%1.10e \n',pi)`
- `fprintf(' x =%6.2f \n',pi)`
- `fprintf(' x =%12.8f y =%12.8f \n',5,exp(5))`
- Note: `\n` is the command for a new line.
- For full information type `>>help fprintf`



Colon (:) Command

- `>>clear; close all; % close figure windows`
- `>>x=0:1:10;`
- `>>A=magic(3)`
- `>>B=A(:,2)`
- `>>C=A(2,:)`
- `>>D=A(2,1:2)`
- `>>E=[A B]`
- `>>F=[A;C]`

- `>>dx=.01;`
- `>>x=.5*dx:dx:10-0.5*dx;`
- `>>y=sin(5*x);`
- `>>length(x)`
- `>>length(y)`
- `>>plot(x,y,'r-')`

MATLAB 也支援複數運算

- `>> z=5+4j` %複數 $z=5+4\sqrt{-1}$ ，通常以 i 或 j 代表單位虛數
z=
5.0000 + 4.0000 i
- `>> z=5+4i` %這也是複數 $z=5+4\sqrt{-1}$
- `>> y=angle(z)` %複數z 的相角
y=
0.6747
- `>> y = real(z)` %複數z 的實部
- `>> y =imag(z)` %複數z 的虛部
- `>> y =conj(z)` %複數z 的共軛複數
- `>> y = z'` %這也是複數z 的共軛複數
y=
5.0000 - 4.0000 i

Euler Identity 尤拉恆等式

■ `>> y = exp(j*pi/6) % $e^{j\theta} = \cos \theta + j \sin \theta$`

y=

0.8660 + 0.5000 i



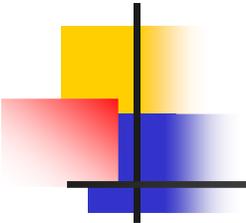
$\cos(\pi/6)$



$\sin(\pi/6)$

向量矩陣的運算

- 有一些函數是特別針對向量而設計
 - `>> y = min(x)` % 向量 X 的極小值
 - `>> y = max(x)` % 向量 X 的極大值
 - `>> y = mean(x)` % 向量 X 的平均值
 - `>> y = sum(x)` % 向量 X 的總和
 - `>> y = sort(x)` % 向量 X 的排序
 - `>> y = sqrt(x)` % 向量 X 的開平方根



sort 指令

- $a = [92, 95, 58, 75, 69, 82]$ ，執行 sort 指令：
- $[b, \text{index}] = \text{sort}(a)$
- 會得到
 - $b = [58, 69, 75, 82, 92, 95]$
 - 及 $\text{index} = [3, 5, 4, 6, 1, 2]$
 - 其中 index 的每個元素代表 b 的每個元素在 a 的位置，
 - 滿足 b 等於 $a(\text{index})$

線上支援

- **help**：用來查詢已知指令的用法
 - `>>help atan2`
- **lookfor**：用來尋找未知的指令。找到所需的指令後，即可用 **help** 進一步找出其用法
 - `help elliptic is no help`
 - `lookfor elliptic`
 - will tell you that you should use
 - `help ellipke`
- **helpwin** 或 **helpdesk**：產生線上支援視窗，其效果和直接點選 MATLAB 命令視窗工作列的圖示是一樣的。
- **doc**：產生特定函數的線上支援。

2-4 程式流程控制

- MATLAB 提供重複迴圈（Loops）及條件判斷（Conditions）等程式流程控制（Flow Control）的指令
 - for 迴圈
For 變數 = 向量
運算式;
end

流程控制

- while 迴圈 (While-loop)

```
while 條件式  
    運算式;  
end
```

- if – else – end

```
if 條件式  
    運算式;  
else  
    運算式;  
end
```

更多內容，[參閱第16章](#)

2-5 m 檔案

- 若要一次執行大量的 MATLAB 指令，可將這些指令存放於一個副檔名為 m 的檔案，並在指令提示號(>>)下鍵入此檔案主檔名即可。

>> pwd % 顯示目前的工作目錄

>> dir 顯示目前工作目錄的內容

>> cd 可改變工作目錄

>> type myTest.m % 顯示 myTest.m 的內容

>> myTest % 執行 myTest.m

- “Save and Run” shortcut key, F5.

```
m file中程式太長可用... 連接  
a=sin(x)*exp(-y)*...  
log(z)+sqrt(b);
```

更多內容，[參閱第15章](#)

2-6 搜尋路徑

- 若要檢視 MATLAB 已設定的搜尋路徑，鍵入 path 指令即可：
>>> path
- 若只要查詢某一特定指令所在的搜尋路徑，可用 which 指令
- 要將目錄加入 MATLAB 的搜尋路徑之最前端，使用 addpath 指令，如
>>> addpath d:\matlabbook
- 要將目錄加入 MATLAB 的搜尋路徑之最末，使用 addpath 指令，如
>>> addpath d:\matlabbook -end
- 要將目錄自 MATLAB 的搜尋路徑移除，使用 rmpath 指令，如
>>> rmpath d:\matlabbook
- 搜尋路徑定義在matlabrc.m 檔案內，可直接修改該檔案

2-7 工作空間與變數的儲存及載入

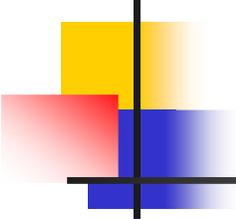
- MATLAB 在進行各種運算時，會將變數儲存在記憶體內，這些儲存變數的記憶體空間稱為基本工作空間（Base Workspace）或簡稱工作空間（Workspace）
 - 若要檢視現存於工作空間（Workspace）的變數，可鍵入 `who`
 - 若要知道這些變數更詳細的資料，可使用 `whos` 指令

永久變數

i 或 j	基本虛數單位
eps	系統的浮點精確度
inf	無限大
nan 或 NaN	非數值 (not a number) 如 0/0
pi	圓周率
realmax	系統所能表示的最大數值
realmin	系統所能表示的最小數值
nargin	函數的輸入引數個數
nargout	函數的輸出引數個數

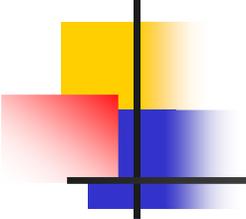
檢視工作空間變數的其他方式

- 使用 `clear` 指令來清除或刪除工作空間內的某一特定或所有變數，以避免記憶體的空置與浪費
 - `clear A`指令清除變數A
 - `clear all`指令清除所有變數
- 不加任何選項（Options）時，`save` 指令會將工作空間內的變數以二進制（Binary）的方式儲存至副檔名為 `mat` 的檔案，`load` 指令可讀取儲存的變數
 - `save`：將工作空間的所有變數以8位元大小儲存到名為 `matlab.mat` 的二進制檔案。`-double` 可改用16位元大小儲存，`-ascii`可改用文字檔案格式儲存。
 - `save filename`：將工作空間所有變數儲存到名為 `filename.mat`的二進制檔案。
 - `save filename x y z`：將變數 `x`、`y`、`z` 儲存到名為 `filename.mat` 的二進制檔案。
 - `load filename`：讀取儲存到名為 `filename.mat`的所有變數



2-8 離開 MATLAB

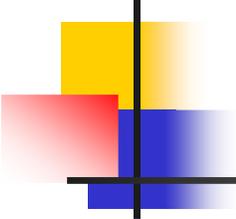
- 在命令視窗內，鍵入 `exit` 指令。
- 在命令視窗內，鍵入 `quit` 指令。
- 直接關閉 MATLAB 的命令視窗。



寫一個程式 findN01.m，求 $n! > \text{realmax}$ 的最小 n 值

- 請問 n 值是多少？此時 $(n-1)!$ 的值又是多少？

```
function findN01
maxN = 1000;
for n=1:maxN
    value = prod(1:n);
    if value>realmax
        break;
    end
end
fprintf('n = %d\n', n);
fprintf('(n-1)! = %d\n', prod(1:n-1));
```



寫一個 myFun01.m 來計算下列函數

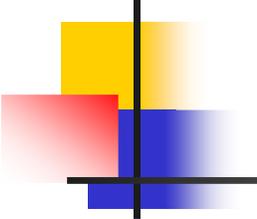
$$y = 0.5 * \exp(x/3) - x * x * \sin(x)$$

其中 x 是函數的輸入， y 是函數的輸出。

你的函數必須能夠處理當 x 是純量或是向量的兩種情況。
此外，請利用下述兩列程式碼來畫出此函數的圖形：

```
x=0:0.1:10;  
plot(x, myFun01(x));
```

```
function y = myFun01(x)  
y = 0.5*exp(x/3)-x.*x.*sin(x);
```



寫一個 piFun01.m 來計算下列級數

$$f(n) = 4*(1 - 1/3 + 1/5 - 1/7 + 1/9 - 1/11 + \dots)$$

其中 n 為函數的輸入，代表上述級數的項數，
級數和 f(n) 則是函數的輸出。當 n 夠大，f(n) 趨近 pi

```
function out = piFun01(n) % approximate pi
total=0;
for i=1:n
    item = (-1)^(i+1)/(2*i-1);
    total = total+item;
end
out = 4*total;
```

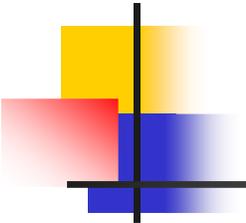
寫一個遞迴函數 fibo.m 來計算 Fibonacci 數列

定義如下： $\text{fibo}(n+2) = \text{fibo}(n+1) + \text{fibo}(n)$

此數列的啓始條件： $\text{fibo}(1) = 0, \text{fibo}(2) = 1$.

使用 tic 和 toc 指令來測量 fibo(25) 的計算時間。

```
function out = fibo(n)
% fibo: Fibonacci number
if n==1
    out=0;
    return;
elseif n==2
    out=1;
    return;
else
    out=fibo(n-1)+fibo(n-2);
end
```



寫一個非遞迴函數 fibo2.m 來計算 Fibonacci 數列

Fibonacci 數列的第 n 項可以直接表示成

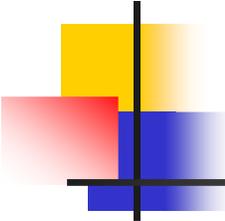
$$\text{fibo2}(n) = \left\{ \left[\frac{1+a}{2} \right]^{(n-1)} - \left[\frac{1-a}{2} \right]^{(n-1)} \right\} / a$$

其中 a 是 5 的平方根。

請計算 $\text{fibo2}(25)$ 的計算時間，並和 $\text{fibo}(25)$ 比較。

```
function out = fibo2(n)
% Fibonacci number using an analytic expression

r1=(1+sqrt(5))/2;
r2=(1-sqrt(5))/2;
out=(r1^(n-1)-r2^(n-1))/sqrt(5);
```



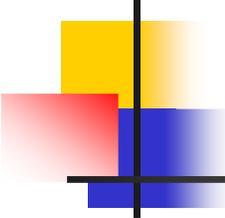
請寫一個函數 `minxy.m`，
其功能是由一個二維矩陣中找出小元素

```
[minValue, minIndex] = minxy(matrix)
```

其中 `matrix` 是一個二維矩陣，`minValue` 則是其元素的最小值，而 `minIndex` 是長度為 2 的正整數向量，代表最小值的索引。換句話說，`matrix(minIndex(1), minIndex(2))` 的值即是 `minValue`。
請測試 `[minValue, minIndex] = minxy(magic(20))` 所傳回來的 `minValue` 和 `minIndex` 各是多少？

```
function [minValue, minIndex] = minxy(matrix)
%Minimum of a 2D matrix
%      Usage: [minValue, minIndex] = minxy(A)
%              minValue: the minimum of the matrix A
%              minIndex: the 2D index of minValue in A

[columnMin, columnMinIndex] = min(matrix);
[minValue, tmp] = min(columnMin);
minIndex = [columnMinIndex(tmp) tmp];
```

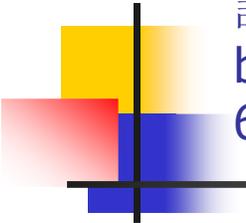


請寫一個函數 ranking01.m ，
輸入為成績向量 x ，輸出則是此成績的排名

```
function out = ranking01(x)
% ranking: Ascending ranking of element of x
% x = [92, 95, 58, 75, 69, 82] 時，ranking01(x) 回傳的排名向量則是
% [2, 1, 6, 4, 5, 3]，代表 92 分是排名第 2，82 分是排名第 3。

[sorted, position]=sort(-x);% 由大到小排列
n=length(x);
rank=1:n;
[junk, index]=sort(position);
out=rank(index);
```

MATLAB 程式設計入門篇：初探MATLAB



請寫一個函數 `sort01.m`，當輸入為 `a` 時，可傳回 `index2`，滿足 `a` 等於 `b(index2)`。以 `a = [92, 95, 58, 75, 69, 82]` 為例，傳回的 `index2` 應該是 `[5, 6, 1, 3, 2, 4]`，顯示由小至大排列的排名。
