

## MATLAB 語言

MATLAB 語言是一種互動性的數學手稿語言，其語法與 C/C++ 類似。它支援包括邏輯 (boolean)、數值 (numeric)、文字 (text)、函數柄 (function handle) 和異質資料容器 (heterogeneous container) 在內的 15 種資料類型，每一種類型都定義為矩陣或陣列的形式 (0 維至任意高維) [14]。

執行 MATLAB 代碼的最簡單方式是在 MATLAB 程式的命令窗口 (Command Window) 的提示符處 ( `>>` ) 輸入代碼，MATLAB 會即時返回操作結果 (如果有的話)。此時，MATLAB 可以看作是一個互動式的數學終端，簡單來說，一個功能強大的「計算機」。MATLAB 代碼同樣可以儲存在一個以 *.m* 為字尾名的文字檔案中，然後在命令窗口或其它函數中直接呼叫。

### 變數與賦值

MATLAB 的變數名字跟許多程式語言一樣，嚴格區分大小寫，例如，*var*、*VAR* 和 *Var* 是三個不同的變數。另外，MATLAB 中變數名字必須以字母為首字母，*3var*、*\_var* 等是非法的變數名。變數由設定運算子 ( `=` ) 定義。MATLAB 是動態檢查的，這意味著變數可以在未定義其類型的情況下賦值並且變數的類型也可以改變，除非將變數看做是符號物件 [15]。變數值可以取自常數，計算中的其他變數的值，或某一函數的輸出。例如：

```
>> x = 17
x =
    17

>> x = 'hat'
x =
```

```
hat

>> x = [3*4, pi/2]
x =
    12.0000    1.5708

>> y = 3*sin(x)
y =
   -1.6097    3.0000
```

## 向量和矩陣

MATLAB的意思是"矩陣實驗室"，因此它提供了許多建立向量、矩陣和多維陣列的便捷的方式。在MATLAB自己的語言中，一個*向量* (vector) 指的是一維 ( $1 \times N$  或  $N \times 1$ ) 矩陣，在其他語言中通常被叫做陣列 (array)。矩陣 (matrix) 通常指的是2-維陣列，例如  $m \times n$  陣列其中  $m$  和  $n$  大於或等於1。多維陣列通常指的是維數大於2的陣列。

需要注意的是，雖然MATLAB用C語言重寫，但是在矩陣儲存方式上卻和FORTRAN保持一致，兩者使用的均為列優先儲存，而非行優先儲存<sup>[c]</sup>。在進行較大的矩陣運算時，這一差別對效能的影響是不應忽略的。

MATLAB提供了定義簡單陣列的簡單方式，使用語法：`初值`：`增量`：`終值`。例如：

```
>> array = 1:2:9
array =
    1    3    5    7    9
```

定義了一個名為 `array` 的變數 (或向已存在的變數 `array` 賦予一個新值)，其值為包含1, 3, 5, 7, 9的一個向量。即陣列由1開始 (初值)，每次以增量2 (增量) 遞增，當達到9 (終值) 時停止。

```
>> array = 1:3:9
array =
    1    4    7
```

在該語法中 增量可被省略 ( 這樣僅剩一個冒號 ) ，這時增量採用預設值1 。

```
>> ari = 1:5
ari =
 1 2 3 4 5
```

由於預設增量為1 ，上面的語句對變數 `ari` 用1, 2, 3, 4, 5的向量賦值 。

在MATLAB內定義矩陣同樣很簡單 ，例如：

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

在MATLAB記憶體取矩陣元素同樣非常簡單 ，例如：

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

>> M(2,2)
ans = 5

>> M(2,:)
ans = 4 5 6

>> M(:,1)
ans = 1
```

```
4
7
>> M(2,2) = 10
M =
    1     2     3
    4    10     6
    7     8     9
```

若要對矩陣內所有元素做同樣的運算，如次方倍，則需要在運算元前面加上"."，例如：

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

>> M.^2
ans =
    1     4     9
   16    25    36
   49    64    81
```

若不在運算元前面加上"."則代表矩陣運算：

```
>> M = [1,2,3;4,5,6;7,8,9]
M^2 = M * M =
   30    36    42
   66    81    96
  102   126   150
```

代數/符號運算

利用MATLAB的符號數學工具箱 ( Symbolic Math Toolbox ) [\[d\]](#)可以進行代數或符號運算，如分解多項式因子和解代數方程式：

```
>> syms x y

>> A=x^3-3*x^2*y+3*x*y^2-y^3;

>> factor(A)
ans =
(x - y)^3
```

把多項式  $A = x^3 - 3x^2y + 3xy^2 - y^3$  簡化成了  $(x - y)^3$ 。

```
>> solve('x^2 - 2*x - 4 = 0')
ans =
1 - 5^(1/2)
5^(1/2) + 1
```

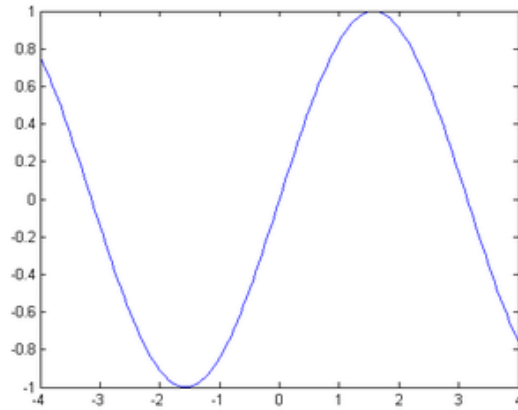
求得方程式  $x^2 - 2x - 4 = 0$  的兩個解為  $1 + \sqrt{5}$  和  $1 - \sqrt{5}$ 。

## 圖形和圖形化使用者介面編程

---

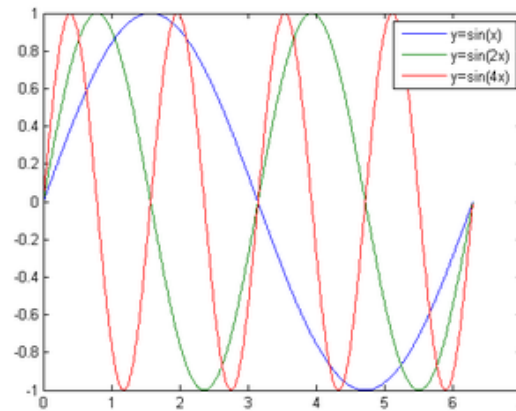
MATLAB中常用的繪圖命令是 *plot*，例如描繪一個在  $[-4,4]$  區間內的正弦函數：

```
>> x = -4:0.05:4;  
>> y = sin(x);  
>> plot(x,y)
```



利用`plot`命令也可以在同一幅圖中描繪多個函數圖形：

```
>> x = 0:.01:2*pi;  
>> y1 = sin(x);  
>> y2 = sin(2*x);  
>> y3 = sin(4*x);  
>> plot(x, [y1; y2; y3])  
>> legend('y=sin(x)', 'y=sin(2x)', 'y=sin(4x)')
```



用MATLAB的畫圖命令可以生成複雜的三維圖形<sup>[e]</sup>：

```
>> t = linspace(0, 2*pi, 512);
>> [u,v] = meshgrid(t);
>> a = -0.4; b = .5; c = .1;
>> n = 3;
>> x =(a*(1 - v/(2*pi)) .* (1+cos(u)) + c) .* cos(n*v);
>> y =(a*(1 - v/(2*pi)) .* (1+cos(u)) + c) .* sin(n*v);
>> z = b*v/(2*pi) + a*(1 - v/(2*pi)) .* sin(u);
>> surf(x,y,z,y)
>> axis off
>> axis equal
>> colormap(hsv(1024))
>> shading interp
>> material shiny
>> lighting phong
>> camlight('left', 'infinite')
>> view([-160 25])
```

