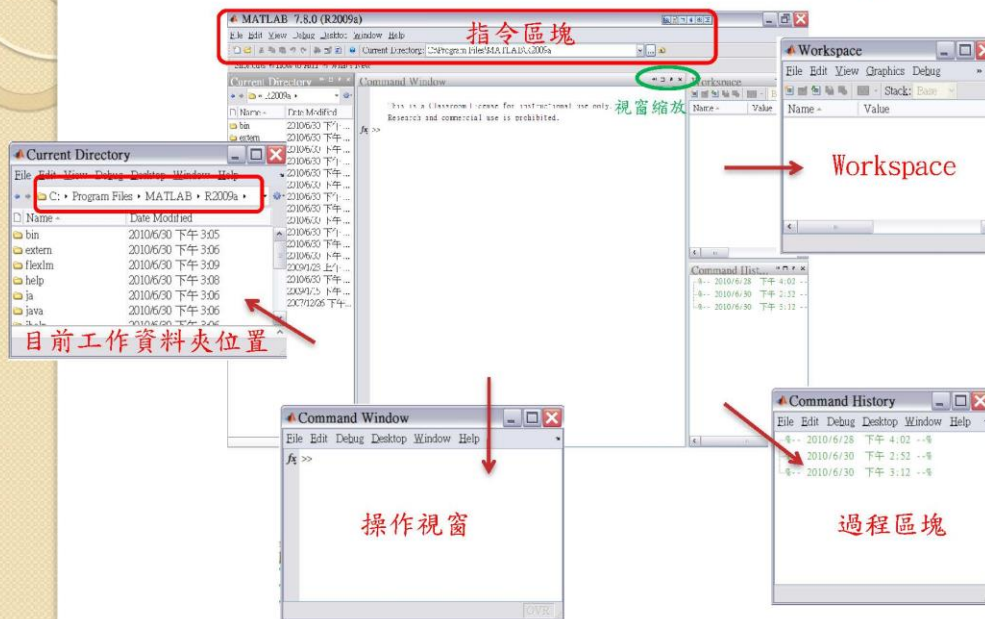


The MATLAB Desktop



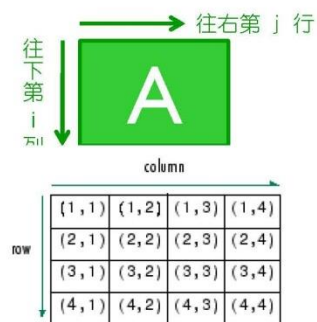
2010/9/6 Design by Pesce @2010

8

代表矩陣元素的變數

□ 說明矩陣(Matrix)位址：

- $A(i,j)$ 代表A的(i,j)位置元素
- $A(i,:)$ 代表A的第i列
- $A(:,n)$ 代表A的第n行
- $A(i,l:k)$ 代表A的第i列之第一行到第k行



2010/9/6 Design by Pesce @2010

22

練習

	1	2	3	4	5
1	4 ¹	10 ⁶	1 ¹¹	6 ¹⁶	2 ²¹
2	8 ²	1.2 ⁷	9 ¹²	4 ¹⁷	25 ²²
3	7.2 ³	5 ⁸	7 ¹³	1 ¹⁸	11 ²³
4	0 ⁴	0.5 ⁹	4 ¹⁴	5 ¹⁹	56 ²⁴
5	23 ⁵	83 ¹⁰	13 ¹⁵	0 ²⁰	10 ²⁵

$a=A(?)$ → (row 3, col 1) ← (row 3, col 5) $b=A(?)$

↑ (row 3, col 2) $c=A(?)$

開啟MATLAB

□什麼是M-file：

若在 Command Window 上面寫程式，按 Enter 後發現寫錯，很難直接修改。

- 例如：把 $err = 5e-15$ 改成 $err = 1e-10$

寫程式最好另外開啟一個檔案，把程式的「原始碼」寫在裡面，這個檔案稱為 **M-file**，此檔案是純文字檔，而且副檔名必須是 **m**。

- 優點：
1. 方便修改。
 2. 不會因為關閉 Matlab 而失去原始碼。

自訂函式(function)

- 在M-file裡，完成你要的函式。
然後在程式的最上方，加入以下語法：
`function 變數名稱=函式名稱(變數名稱)`
 - EX: `function n = apple(x)`
 - 等號的左邊是要output的數據的變數名稱
 - 等號的右邊是要 input的數據的變數名稱
 - 輸入參數是從呼叫者 (caller) 那裡傳過來的
- 儲存函式：請將M-file和函式取同樣的名字，這樣將來比較容易管理。

Structure of a Function M-File

□ 語法

```
function [out1, out2, ...] = fun_name(in1, in2, ...)
```

Function name (same as filename.m)

Online help

```
% Program 12.4 Normalized Simultaneous Iteration  
% Computes eigenvalues/vectors of symmetric matrix  
% Input: matrix A, number of steps k  
% Output: eigenvalues lam and eigenvector matrix Q
```

MATLAB
code

```
function [lam,Q]=nsi(A,k)  
[m,n]=size(A);  
Q = eye(m,m);  
for j = 1:k  
    [Q,R] = qr(A*Q);  
end  
lam=diag(Q'*A*Q);
```

■ MATLAB 用於求解常微分方程式的指令：

指令	方法	適用ODE類別
ode45	Explicit Runge-Kutta (4, 5) pair of Dormand-Prince	Nonstiff ODE
ode23	Explicit Runge-Kutta (2, 3) pair of Bogacki and Shampine	Nonstiff ODE
ode113	Variable order Adams-Bashforth-Moulton PECE solver	Nonstiff ODE
ode15s	Numerical differentiation formulas (NDF5)	Stiff ODE
ode23s	Modified Rosenbrock formula of order 2	Stiff ODE
ode23t	Trapezoidal rule with a "free" interpolant	Stiff ODE
ode23tb	Implicit Runge-Kutta formula with a backward differentiation formula of order two	Stiff ODE

■ 指令項目繁多，最主要可分兩大類

■ 適用於 Nonstiff 系統

- 一般的常微分方程式都是 Nonstiff 系統
- 直接選用 ode45、ode23 或 ode113 來求解

■ 適用 Stiff 系統

- 速率（即微分值）差異相常大
- 使用一般的 ode45、ode23 或 ode113 來求解，可能會使得積分的步長（Step Sizes）變得很小，以便降低積分誤差至可容忍範圍以內，會導致計算時間過長
- 專門對付 Stiff 系統的指令，例如 ode15s、ode23s、ode23t 及 ode 23tb

以上資料從

na.math.fju.edu.tw/輔大張康教授網頁截取

張智星教授的網頁 <http://www.cs.nthu.edu.tw/~jang>

www.mirlab.org/jang/books/matlabProgramming4guru/slide/11-常微分方程式.ppt

執行 ODE 的流程

1. 先造一個 ” function”

以 vdp1.m 為例

```
function dydt = vdp1(t,y)
dydt = [    y(2)
          (1-y(1)^2)*y(2)-y(1)  ];
```

EX: 改寫 vdp1.m 記成 vdp.m 使得 mu 為參數

2. 以下三種方式可擇一執行

(1) 可以簡單的在”操作介面”執行

```
>> ode45(@vdp1,[0 100 ],[ 3 3]);
```

時間 起始點

(2) 在”操作介面”執行，把執行ode45的解儲存另外畫圖

```
>> [t y]=ode45(@vdp1,[0 100 ],[ 3 ;3]);
>> plot(t, y(:,1), t, y(:,2),'o')
```

(3) 把執行的步驟存成 **m-file**，以利更改

```
clear all
global mu

mu=5;
X0 = [ 2 7 ];
[ t, X ] = ode45(@vdp,[0 100], X0);

% plot( t,[X(:,1),X(:,2)]);
% legend('x(t)', 'x''(t)')
% xlabel('Time (t)')
% ylabel('Solution x(t) and x''(t)')

subplot(2,1,1); plot( t,[X(:,1),X(:,2)]);
legend('x(t)', 'x''(t)')
xlabel('Time (t)')
ylabel('Solution x(t) and x''(t)')

subplot(2,1,2); plot( X(:,1),X(:,2));
xlabel('x (t)')
ylabel(' x''(t)')

disp(X(end, :));
```

在例子中，我們取 $\mu=5$ ，此為 **nonstiff ODE**

若 $\mu=1000$ ，此為 **stiff ODE**，須改用 **ode15s**

```
[ t, X ] = ode15s(@vdp,[0 100], X0);
```